

ASG-SmartEdit™

User's Guide

Version 7.0

Publication Number: SET0200-70

Publication Date: February 2003

The information contained herein is the confidential and proprietary information of Allen Systems Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Allen Systems Group, Inc.

©2003 Allen Systems Group, Inc. All rights reserved.

All names and products contained herein are the trademarks or registered trademarks of their respective holders.

Contents

Preface	ix
About this Publication	x
Related Publications	xi
ASG-Existing Systems Workbench (ASG-ESW)	xii
Invoking ESW Products	xv
ESW Product Integration	xvi
Example 1	xvii
Example 2	xviii
Publication Conventions	xx
ASG Customer Support	xx
Intelligent Support Portal (ISP)	xxi
Telephone Support	xxi
ASG Documentation/Product Enhancements	xxiii
 1 Introduction	 1
Introduction	1
SmartEdit Overview	1
Facilities	2
 2 Product Overview	 5
User Interface	5
Using the Action Bar	6
ISPF 4.1 through 4.8	6
Actions	7
Pull-downs	8
Pop-ups	9
Screens	10

Concepts	11
Screen Messages	11
Sets	12
Subsets.....	12
Targets.....	15
PF Key Values.....	18
 3 Getting Started	 21
Introduction	22
Activating SmartEdit	23
Defining Options - COPY/Include Libraries	24
Loading the COPY List.....	29
Saving the COPY List.....	29
Extracting a COPY	30
Defining Options - Product Parameters	31
Using Edit Entry Screens	32
ISPF/PDS Edit - Entry Screen.....	33
Panvalet Edit - Entry Screen	33
Librarian Entry Screen	34
Editor Screen	34
ISPF 4.1 through 4.8	35
Tracing Program Control Flow	35
Following the Flow of Program Logic	36
Following Program Flow	37
Returning to Previous Branch Locations	39
Viewing Copybooks In-context	40
Adding a Copybook to the Display.....	40
Removing the Copybook from the Display	42
Checking Program Syntax	43
Checking for Syntax Errors.....	44
Locating Errors from the Check Function	45
Using the COBOL Intelligent Search Feature	46
COBOL Intelligent Search Description	46
Finding Data Relationships and References	47
Finding the Impact of Changing a Data Field Size.....	48
Determining the Impact of Changing the Usage of a Data Item.....	50
Finding Subsets of Related COBOL Verbs	51
COBOL Program Understanding Features	53
Understanding Program Structure.....	53
Revealing Program Structure Using the ZOOMIN Function	55
Removing Parts of the Program Structure from the Display	57

Understanding Program Flow of Control	57
Viewing Complete Paragraph Cross-references	60
Gaining High-level Program Understanding	62
Viewing Complete Program	65
Viewing Source Statements	66
Removing Source Statements from the Display	67
Switching Between the Editor and Tree View	68
Using the Intelligent Search within Tree View	70
Ending Tree View	72
Defining Character Substitution Name	72
Troubleshooting	73
 4 File	 77
File Pull-down	77
Actions	78
File Pull-down in Tree View	79
File - Display Logic Segment Entry Pop-up	79
File - Program Name List Pop-up	80
File - Select Logic Segment Pop-up	82
Displaying Logic Segment	83
File - Generated Code Update Facility Pop-up	83
File - Pseudo Code Program Selection	85
 5 View	 87
View Pull-down	88
View Pull-down in Tree View	90
View - Tree View Request Pop-up	91
Tree View Screen	93
Tree View Generated Tags	94
Primary Commands	95
Line Commands	97
View - Paragraph Cross Reference Request Pop-up	98
View - Paragraph Cross Reference Pop-up	99
Data Item Offset and Length Pop-up	102
View - Reset Request Pop-up	103
View - Exclude Request Pop-up	104
View - Levels Pop-up	105
View - Scroll Request Pop-up	106

6 Search	107
Search Pull-down	108
Search - Data Name Pop-up	109
Search - Label Name Pop-up	113
Search - Pattern String Pop-up	116
Search - COBOL Subset Name Pop-up	120
Search - Program Name Pop-up	123
Search - Line Range Pop-up	126
Search - Any/Unknown Type Pop-up	129
Search - Branch Request Pop-up	132
Selection List Pop-ups	134
Search - IN Clause Pop-up	135
7 Check	137
Check Pull-down	137
Syntax Checking Process	139
Check - Compiler Options Pop-up	140
Check - Preprocessors Pop-up	143
Check - Modules and Libraries Pop-up	144
8 List	147
List Pull-down	148
List - CALL Statements Pop-up	149
List - Equates Pop-up	150
List - Perform Range Names Pop-up	151
List - Program/Subprogram Names Pop-up	152
List - COBOL Subset Names Pop-up	153
9 Options	155
Options Pull-down	156
Options - Product Parameters Pop-up	157
DB2 Parameter Definition Pop-up	158
IDMS Parameter Definition Pop-up	159

IDMS Load Module Libraries Pop-up	161
IDMS Local Mode Dictionaries Pop-up	162
DATACOM Parameter Definition Pop-up	163
Options - Product Allocations Pop-up	164
Options - Log/List/Punch Definition Pop-up	165
Customizing Log/List/Punch Dataset Names	167
Options - Processing Modes Pop-up	169
Options - Preprocessors Pop-up	172
Options - Define Preprocessor Group Pop-up	173
Options - Equate Pop-up	174
10 Help	175
Help Pull-down	175
Help - Specific ASG Command Pop-up	177
Help - Specific ASG Message Number Pop-up	178
Help - About Pop-up	179
11 Commands	181
Introduction	183
Command Processing	184
Recalling/Repeating Commands	184
Cursor Position	185
Cursor Substitution Character	186
Command Diagrams	186
& (Retain) Command	189
ACTION Command	190
ALLOCDEF Command	191
BRANCH Command	192
CHECK Command	194
COBEDIT Command	197
CUA Command	198
DATAMAP Command	199
DISPLAY Command	200
EDITCOPY Command	201

EDITPGM Command	203
EQUATE Command	204
EXCLUDE Command	206
FINDXTND Command	212
HELP Command	222
HIGH Command	223
JUMP Command	229
LEVELS Command	230
LIST Command	231
List Menu Pop-up	232
LPRINT Command	233
LPUNCH Command	240
PARMDEF Command	246
PREF Command	247
PREPROC Command	250
PRINTLOG Command	251
PRINTLST Command	252
PRODLVL Command	253
RECALL Command	254
REDO Command	257
REFRESH Command	258
REPEAT Command	259
RESET Command	260
RFIND Command	261
RHIGH Command	262
RPREF Command	263
RSCROLL Command	264
RTREEVW Command	265
SCROLL Command	266
SET Command	272
Generated Command Pop-up	273
SWITCH Command	274

TREEVIEW Command	275
UPDATE Command	277
ZOOM COPY Command	279
ZOOM DEF Command	280
ZOOMIN/ZOOMOUT Commands	281
Line Commands	283
BR (Branch) Line Command	284
DM (DATAMAP) Line Command	285
EC (EDITCOPY) Line Command	286
EP (EDITPGM) Line Command	287
F (First) Line Command	288
J (Jump) Line Command	289
L (Last) Line Command	290
S (Show) Line Command	291
SS (Show Block) Line Command	292
X (Exclude) Line Command	293
XX (Exclude Block) Line Command	294
ZD (Zoom Def) Line Command	295
ZI (Zoom In) Line Command	296
ZO (Zoom Out) Line Command	297
12 Command Quick Reference	299
13 Help Facility	303
Introduction	303
Help Navigational Commands	305
Screen and Pop-up Help	306
Command Help	307
General Information	309
Specific Information	310
Help Abends	311
Help Messages	312
Printing Messages	313
Glossary	317
Index	323

Preface

This *ASG-SmartEdit User's Guide* provides in-depth information about using ASG-SmartEdit (herein called SmartEdit). SmartEdit is a COBOL intelligent editor that increases the speed and accuracy of modifying COBOL code using the ISPF/PDF editor. SmartEdit assists COBOL programmers in understanding:

- Program structure
- How related data fields affect the logic of the program
- The impact of syntax modifications on program logic
- Execution paths of a COBOL program

Clear, easy-to-use command extensions of the standard ISPF/PDF editor facilities provide a powerful interactive system to help automate the process of understanding and modifying COBOL programs.

The *ASG-SmartEdit User's Guide* is written primarily for programmers who maintain and design COBOL programs.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ASG product. Telephone technical support is available around the world, 24 hours a day, 7 days a week.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on any ASG product.

About this Publication

This publication consists of these chapters:

- [Chapter 1, "Introduction,"](#) contains an overview of ASG-ESW and SmartEdit, which is used to extend the power of the ISPF environment to enhance and modify COBOL programs efficiently by saving time and improving the logic quality.
- [Chapter 2, "Product Overview,"](#) contains an overview of SmartEdit, which includes Common User Access (CUA) screens, action bars, pull-downs, and pop-ups that are designed to provide easy access to all of the product features.
- [Chapter 3, "Getting Started,"](#) contains information about SmartEdit that is intended primarily for new users of the product. This information includes how to invoke SmartEdit and how to use specific menu items and associated pull-downs.
- [Chapter 4, "File,"](#) contains specific information about the File pull-down, which is used to save, cancel, or exit a program, and to exit SmartEdit. The File pull-down is also used to access the Display and Update functions.
- [Chapter 5, "View,"](#) contains specific information about the View pull-down, which is used to select the method to view a program, view transfers of control between paragraphs, display or exclude source according to the program hierarchical structure, change the level of information viewed, display the length and offset of data items contained within a selected data item, reset the display, and exclude source from the display.
- [Chapter 6, "Search,"](#) contains specific information about the Search pull-down, which is used to conduct a COBOL intelligent search of the source code for one or all occurrences of a specified target.
- [Chapter 7, "Check,"](#) contains specific information about the Check pull-down, which provides an online syntax check mechanism for the COBOL source currently being edited.
- [Chapter 8, "List,"](#) contains specific information about the List pull-down, which is used to access pop-ups that list information about calls, equates, performs, programs, and subsets.
- [Chapter 9, "Options,"](#) contains specific information about the Options pull-down, which is used to customize your SmartEdit environment by setting certain parameters and options, setting processing modes, defining equates, or refreshing the display.
- [Chapter 10, "Help,"](#) contains specific information about the Help pull-down, which is used to access the online help facility.

- [Chapter 11, "Commands,"](#) contains information about the commands used by SmartEdit, which accepts primary and line commands entered in the same manner as ISPF commands. Primary commands are entered on screens and pop-ups that contain a command input area, and line commands are entered in the prefix area over the line numbers. SmartEdit supports all ISPF system commands on the appropriate screens.
- [Chapter 12, "Command Quick Reference,"](#) contains specific information about SmartEdit primary commands, including command abbreviations, PF key assignments, and command descriptions.
- [Chapter 13, "Help Facility,"](#) contains a description of the comprehensive and context-sensitive Help facilities that are provided to answer most questions online. The Help Tutorial contains help information about several subjects, such as screens, pop-ups, reports, commands, messages, and abends. The Help Tutorial also includes a Table of Contents that describes each major SmartEdit function, and a comprehensive Index for viewing specific information.

Related Publications

The documentation library for ASG-SmartEdit consists of these publications (where *nn* represents the product version number):

- The *ASG-Center Installation Guide* (CNX0300-*nn*) contains installation and maintenance information for ASG-Center, the common set of libraries shared by the Existing Systems Workbench suite of products. ASG-Center must be installed before installing SmartEdit.
- *ASG-ESW Enhancement Summary* (ESW1000-*nn*) highlights the new functionality for this release.
- The *ASG-SmartEdit Installation Guide* (SET0300-*nn*) provides information about the installation and maintenance of ASG-SmartEdit.
- *ASG-SmartEdit Quick Reference Guide* (SET0600-*nn*) summarizes the syntax and usage of ASG-SmartEdit commands.
- The *ASG-SmartEdit User's Guide* (SET0200-*nn*) describes instructions and commands for using ASG-SmartEdit.

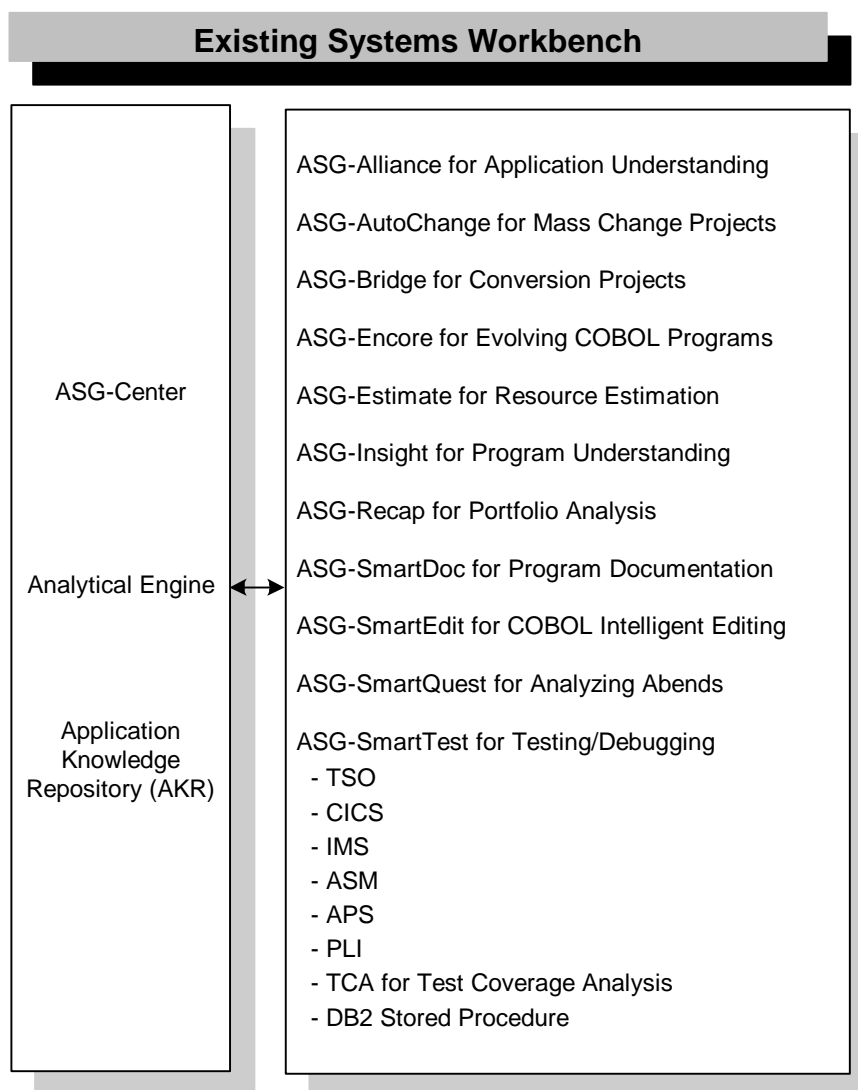
Note: _____

To obtain a specific version of a publication, contact ASG Customer Support.

ASG-Existing Systems Workbench (ASG-ESW)

ASG-ESW (herein called ESW) is an integrated suite of components designed to assist organizations in enhancing, redeveloping, or re-engineering their existing systems. ESW products use the Application Knowledge Repository (AKR) to store source program analysis information generated by the Analytical Engine. [Figure 1](#) represents the components of ESW.

Figure 1 • ASG Existing Systems Workbench



This table contains the name and description of each ESW component:

ESW Product	Herein Called	Description
ASG-Alliance	Alliance	The application understanding component that is used by IT professionals to conduct an analysis of every application in their environment. Alliance supports the analysis and assessment of the impact of change requests upon an entire application. Alliance allows the programmer/analyst to accurately perform application analysis tasks in a fraction of the time it would take to perform these tasks without an automated analysis tool. The impact analysis from Alliance provides application management with additional information for use in determining the resources required for application changes.
ASG-AutoChange	AutoChange	The COBOL code change tool that makes conversion teams more productive by enabling quick and safe changes to be made to large quantities of code. AutoChange is an interactive tool that guides the user through the process of making source code changes.
ASG-Bridge	Bridge	The bridging product that enables field expansion for program source code, without being required to simultaneously expand the fields in files or databases. Because programs are converted in smaller groups, or on a one-by-one basis, and do not require file conversion, testing during the conversion process is simpler and more thorough.
ASG-Center	Center	The common platform for all ESW products. Center provides the common Analytical Engine to analyze the source program and store this information in the AKR. This common platform provides a homogeneous environment for all ESW products to work synergistically.

ESW Product	Herein Called	Description
ASG-Encore	Encore	The program re-engineering component for COBOL programs. Encore includes analysis facilities and allows you to extract code based on the most frequently used re-engineering criteria. The code generation facilities allow you to use the results of the extract to generate a standalone program, a callable module, a complement module, and a CICS server. Prior to code generation, you can view and modify the extracted Logic Segment using the COBOL editor.
ASG-Estimate	Estimate	The resource estimation tool that enables the user to define the scope, determine the impact, and estimate the cost of code conversion for COBOL, Assembler, and PL/I programs. Estimate locates selected data items across an application and determines how they are used (moves, arithmetic operations, and compares). Time and cost factors are applied to these counts, generating cost and personnel resource estimates.
ASG-Insight	Insight	The program understanding component for COBOL programs. Insight allows programmers to expose program structure, identify data flow, find program anomalies, and trace logic paths. It also has automated procedures to assist in debugging program abends, changing a computation, and resolving incorrect program output values.
ASG-Recap	Recap	The portfolio analysis component that evaluates COBOL applications. Recap reports provide function point analysis and metrics information, program quality assessments, intra-application and inter-application comparisons and summaries, and historical reporting of function point and metrics information. The portfolio analysis information can also be viewed interactively or exported to a database, spreadsheet, or graphics package.
ASG-SmartDoc	SmartDoc	The program documentation component for COBOL programs. SmartDoc reports contain control and data flow information, an annotated source listing, structure charts, program summary reports, exception reports for program anomalies, and software metrics.

ESW Product	Herein Called	Description
ASG-SmartEdit	SmartEdit	The COBOL editing component that can be activated automatically when the ISPF/PDF Editor is invoked. SmartEdit provides comprehensive searching, inline copybook display, and syntax checking. SmartEdit allows you to include an additional preprocessor (for example, the APS generator) during syntax checking. SmartEdit supports all versions of IBM COBOL, CICS, SQL, and CA-IDMS.
ASG-SmartQuest	SmartQuest	The diagnostic tool for analyzing batch and CICS transaction abends. SmartQuest has been designed to make the maximum use of simple point-and-shoot techniques to enable fast and easy navigation through any data dump.
ASG-SmartTest	SmartTest	The testing/debugging component for COBOL, PL/I, Assembler, and APS programs in the TSO, MVS Batch, CICS (including file services), and IMS environments. SmartTest features include program analysis commands, execution control, intelligent breakpoints, test coverage, pseudo code with COBOL source update, batch connect, disassembled object code support, and full screen memory display.

Invoking ESW Products

The method you use to invoke an ESW product depends on your system setup. If you need assistance to activate a product, see your systems administrator. If your site starts a product directly, use the ISPF selection or CLIST as indicated by your systems administrator. If your site uses the ESW screen to start a product, initiate the ESW screen using the ISPF selection or CLIST as indicated by your systems administrator and then typing in the product command on the command line.

The product names can also vary depending on whether you access a product directly or through ESW. See ["ESW Product Integration" on page xvi](#) for more information about using ESW.

To initialize ESW products from the main ESW screen, select the appropriate option on the action bar pull-downs or type the product shortcut on the command line.

Product Name (ESW Name)	Shortcut	ESW Pull-down Options
Alliance (Application Understanding)	AL	Understand ► Application
AutoChange (Conversion Set)	CC	Change ► Conversion Set
Bridge	BR	Change ► ASG-Bridge
Encore (Program Re-engineering)	EN	Re-engineer ► Program
Estimate	ES	Measure ► ASG-Estimate
Insight (Program Understanding)	IN	Understand ► Program
Recap (Portfolio Analysis)	RC	Measure ► Portfolio
SmartDoc (Program Documentation)	DC	Document ► Program
SmartEdit	SE	Change ► Program Or Change ► Program with Options
SmartQuest	SQV	Understand ► Abend/Dump
SmartTest (Testing/Debugging)	ST	Test ► Module/Transaction

ESW Product Integration

Because ESW is an integrated suite of products, you are able to access individual ESW products directly, or through the main ESW screen. As a result, different fields, values, action bar options, and pull-down options display on a screen or pop-up depending on how you accessed the screen or pop-up.

Certain ESW products also contain functionality that interfaces with other ESW products. Using SmartTest as an example, if Alliance is installed, SmartTest provides a dynamic link to Alliance that can be used to display program analysis information. If Insight is installed and specified during the analyze, the Insight program analysis functions are automatically available for viewing logic/data relationships and execution path. For example, the Scratchpad option is available on the Options pull-down if you have Insight installed.

Access to these integrated products requires only that they be installed and executed in the same libraries.

Example 1

[Figure 2](#) shows the Encore Primary screen that displays when you access Encore directly.

The Encore Primary screen contains these eight action bar menu items: File, View, Extract, Generate, Search, List, Options, and Help.

Figure 2 • Encore Primary Screen

[illegible]

Figure 3 shows the Encore Primary screen that displays when you access Encore through ESW by selecting Re-engineer ► Program from the ESW action bar menu. Notice that the Primary screen name changes to ASG-ESW - Program Re-engineering when you enter Encore through ESW. Also, the Logic menu item displays if Insight is installed.

Figure 3 • ESW Encore Primary Screen

[illegible]

Example 2

[Figure 4](#) shows the File - Analyze Submit pop-up that displays when you access SmartTest directly. [Figure 5 on page xix](#) shows the File - Analyze Submit pop-up that displays when you access SmartTest through ESW.

Figure 4 • File - Analyze Submit Screen

```

File - Analyze Submit
Command ==> _____
           E - Edit JCL                      S - Submit JCL

Compile and link JCL (PDS or sequential):
  Data set name  _____

Analyze features (Y/N):
  ASG-SmartTest: Y   Extended Analysis: N

AKR data set name _____
AKR program name  NEWDEMO                (if overriding PROGRAM-ID)

Analyze options:
  _____
  _____
  _____

Compile? (Y/N) . . . . . Y   (Y if needed by features)
Link load module reusable? (Y/N) Y

```

The actions shown on these screens can also vary. For example, the D - Doc Options action is only available on the File Prepare Program screen (or File - Analyze Submit screen) if SmartDoc is installed on your system. In [Figure 4 on page xviii](#), the Doc Options action is not displayed.

Figure 5 • ASG-ESW - Prepare Program Screen (accessed through ESW)

```

ASG-ESW - Prepare Program
Command ==> -----
          E - Edit JCL      S - Submit JCL      D - Doc Options

Compile and link JCL (PDS or sequential):
Data set name -----

Analyze features (Y/N):
  Understand: N  Test: Y  Extended Analysis: N  Document: N
  Re-engineer: N  Abend/Dump: N
AKR data set name -----
AKR program name  NEUDEMO ----- (if overriding PROGRAM-ID)

Analyze options:
-----
-----

Compile? (Y/N) . . . . . Y      (Y if needed by features)
Link load module reusable? (Y/N) Y      (Test and Abend/Dump only)

```

Notice that the Analyze features field in [Figure 5](#) lists additional ESW products than shown on [Figure 4 on page xviii](#). This field is automatically customized to contain the ESW products you have installed on your system. These are the names of the analyze types:

Analyze Type	Analyze Type (ESW)
ASG-Encore	Re-engineer
ASG-Insight	Understand
ASG-SmartDoc	Document
ASG-SmartQuest	Abend/Dump
ASG-SmartTest	Test
Extended Analysis (ASG-SmartTest with Insight installed)	Extended Analysis

Publication Conventions

ASG uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
<i>lowercase italic monospace</i>	Information that you provide according to your particular situation. For example, you would replace <i>filename</i> with the actual name of the file.
Monospace	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.
Vertical Separator Bar () with underline	Options available with the default value underlined (e.g., Y <u>N</u>).
<u>Underline</u>	Denotes a cursor-selectable field or line.

ASG Customer Support

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours.

ASG Third-party Support. ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Intelligent Support Portal (ISP)

Online product support is available at: <http://www.asg.com/support/support.asp> via the ASG Intelligent Support Portal (ISP). Your logon information for ISP online support is:

Customer ID = *NNNNNNNNNN*

Password = *XXXXXXXXXX*

where:

NNNNNNNNNN is your customer ID supplied by ASG Product Distribution.

XXXXXXXXXX is your unique password supplied by ASG Product Distribution.

The *ASG-Intelligent Support Portal User's Guide* provides instructions on how to use the ISP and is located on the ASG Support web page.

Telephone Support

To expedite response time, please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely as displayed
- A description of the specific steps that immediately preceded the problem
- Verify whether you received an ASG Service Pack or cumulative service tape for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack. You can access the latest software corrections and Service Packs via the ISP.
- The severity code (ASG Customer Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

The Americas

	Phone	Fax	E-mail
United States and Canada	800.354.3578	1.703.464.4901	support@asg.com

Europe, Middle East, and Africa (EMEA)

During normal business hours, we recommend that you call the Central Support number first (except in South Africa).

	Phone	Fax	E-mail
Central Support	00.800.3544.3578	44.1727.812018	support.emea@asg.com
English	44.1727.736305	44.1727.812018	support.uk@asg.com
French	33.141.028590	33.141.028589	support.fr@asg.com
German	49.89.45716.200	49.89.45716.400	support.de@asg.com
Italian	39.0290450025		support.it@asg.com
Dutch	31.30.241.6133		support.nl@asg.com
Spanish	34.913.523.800	34.917.156.961	support.es@asg.com
South Africa	800.201.423		support.sa@asg.com

Asia Pacific (APAC)

	Phone	Fax	E-mail
Central Support	61.3.9645.8500	61.3.9645.8077	support.au@asg.com
Australia	800.637.947	61.3.9645.8077	support.au@asg.com
Hong Kong	800.96.2800		support.hk@asg.com
Japan	81.3.5326.3684	81.3.5326.3001	support.au@asg.com
Singapore	65.224.3080	65.224.8516	support.sg@asg.com

All Other Countries (Also for any non-working numbers)

	Phone	Fax	E-mail
All other countries	1.239.435.2201		support@asg.com

If you receive a voice mail message, follow the instructions to report a production-down or critical problem. Leave a detailed message including your name and phone number. An ASG Customer Support representative will be paged and will return your call as soon as possible. Please have available the information described previously when the ASG Customer Support representative contacts you.

ASG Documentation/Product Enhancements

Submit all product and documentation suggestions to ASG's product management team at <http://www.asg.com/asp/emailproductsuggestions.asp>.

If you do not have access to the web, FAX your suggestions to product management at (239) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

1

Introduction

This chapter introduces SmartEdit and contains these sections:

Section	Page
Introduction	1
SmartEdit Overview	1
Facilities	2

Introduction

SmartEdit is the COBOL-editing component of ESW and is activated automatically when the ISPF/PDF editor is invoked. It provides comprehensive searching, inline copybook display, and syntax checking. SmartEdit allows you to include an additional preprocessor during syntax checking (i.e., the APS generator). SmartEdit supports most versions of IBM COBOL, CICS, SQL, and CA-IDMS.

SmartEdit Overview

SmartEdit extends the power of the ISPF environment to enhance and modify COBOL programs efficiently by saving time and improving the logic quality. All command and function facilities are 100 percent compatible with the ISPF/PDF editor requiring little or no learning curve.

SmartEdit features Common User Access (CUA) screens, pull-down menus, and pop-ups that are designed to provide easy access to all of the product features.

CUA screens feature action bars, pull-down menus, and pop-ups. An action bar is the line of keywords that display at the top of a screen. Each keyword represents a category of actions that can be performed on that screen. To view the list of these actions on a pull-down menu, move the cursor to the desired keyword and press Enter. To make a selection, enter the number that corresponds to the desired action and press Enter, or move the cursor to the desired action and press Enter. The result may be a generated command or a pop-up.

A pop-up is a window that is superimposed on your screen to allow entry of information for the requested action. Type the desired data and options and press Enter to execute the action.

For more information about the CUA interface, see ["Product Overview" on page 5](#).

Facilities

SmartEdit provides facilities that are easily accessed using pull-down menu options and pop-ups, or by using the familiar command extensions within ISPF.

- Convenient COBOL-sensitive facilities reduce the amount of time it takes to complete basic tasks, such as organizing and viewing the source code. These functions are included:
 - A cursor-sensitive BRANCH facility that automatically locates code for all COBOL transfer of control verbs, such as PERFORMs and GO TOs.
 - A syntax checking facility that provides online syntax checking of COBOL code. Errors are highlighted and explanatory messages are provided below the line where the error is encountered.
 - A Zoom In facility that displays program hierarchy and expands copybooks inline and in-context.
 - A Zoom Out facility reverses the effects of the Zoom In action or the ZOOMIN command.
 - A Zoom Definition facility that allows the user to see data definitions in the procedure division.
 - A Zoom All Copies facility that expands all copy directives simultaneously.
- The Search facility (FINDXTND command) is an automated COBOL-sensitive search that performs these functions:
 - Displays all references to a data item by tagging and highlighting the references (REF), uses (USE), modifications (MOD), and definitions (DEF) of all related data items.
 - Finds all related data (i.e., aliases, renames, redefines) for a specified data field.

- Locates COBOL verbs and subsets, such as COBOL II, CICS, IO, and Exits.
- Provides an overview of the program structure such as divisions, structures, and perform ranges.
- Finds and tags results within expanded or unexpanded copy members.
- COBOL understanding features that identify execution patterns between modules, paragraphs, and statements. These features are included:
 - A Tree View facility that exposes the logical structure of a program, such as perform ranges, calls, conditionals, and GO TOs.
 - A View - Paragraph Cross Reference pop-up that shows the program logic at the paragraph level. You can access this pop-up by selecting View ► Paragraph X-Ref or by typing PREF on the command line.

2

Product Overview

This chapter provides a product overview of SmartEdit and contains these sections:

Section	Page
User Interface	5
Using the Action Bar	6
Concepts	11

User Interface

SmartEdit features Common User Access (CUA) screens, action bars, pull-downs, and pop-ups that are designed to provide easy access to all of the product features.

An action bar is the line of keywords that displays at the top of a screen. Each keyword represents a category of actions that may be performed on that screen. Select an action by moving the cursor to the desired keyword and pressing Enter. A pull-down menu displays listing available actions.

A pop-up is a window that displays as the result of selecting an item on a pull-down or another pop-up, or as the result of entering certain commands. It is superimposed on your screen to allow entry of information for the requested action. To process the information, type the desired data and/or option, and follow the instructions on the pop-up.

Use the END command (default PF3/15) to exit a pull-down or pop-up without processing any actions.

Using the Action Bar

The Action bar displays on all SmartEdit screens. On a few pop-ups, a shortened action bar displays with actions specific to that pop-up. The shortened action bar contains fewer choices than the regular action bar and some of the pull-downs on the shortened action bar contain fewer actions.

Select an action by using the Home and Tab keys to move the cursor to the desired keyword and pressing Enter.

If the cursor is on a full screen, pressing Home moves the cursor to the first action on the action bar. If the cursor is on the action bar when PF3/15 is pressed, the edit session is ended.

ISPF 4.1 through 4.8

If you are using ISPF 4.1 through 4.8, the operation of the Home and Tab keys is controlled by ISPF settings.

To enable the use of the Home and Tab keys

- 1 Type SETTINGS.in in the command input area to display the ISPF Settings screen.
- 2 Type a / in the selection field beside Tab to action bar choices.
- 3 Press PF3 to return to the SmartEdit screen.

After performing these steps, use the Home and Tab keys to access the action bar as described in [Using the Action Bar](#).

[Figure 6](#) shows the SmartEdit action bar on the Editor screen. This action bar designates the primary functional organization of SmartEdit.

Figure 6 • Action Bar Format (Editor Screen)

```

File View Search Check List Options Help
-----
ASG-ESW - COBOL EDIT - VIAINST.CE50D001.CNTL(VIAFDEMO) - 5 COLUMNS 00001 00072
Command ==> _____ $scroll ==> PAGE
***** Top of Data *****
-Warning- The UNDO command is not available until you change
your edit profile using the command RECOVERY ON.
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. VIAFDEMO.
000300 000300 AUTHOR. WRITTEN BY ASG AT LANGLVL 2.
000400 000400 ENVIRONMENT DIVISION.
000500 000500 CONFIGURATION SECTION.
000600 000600 SOURCE-COMPUTER. IBM-370.
000700 000700 OBJECT-COMPUTER. IBM-370.
000800 000800 INPUT-OUTPUT SECTION.
000900 000900 FILE-CONTROL.
001000 001000     SELECT MASTERIN ASSIGN TO S-MASTERIN.
001100 001100     SELECT MASTER-RPT ASSIGN TO S-MREPORT.
001200 001200 DATA DIVISION.
001300 001300 FILE SECTION.
001400 001400 FD MASTERIN
001500 001500     RECORDING MODE IS F
001600 001600     BLOCK CONTAINS 0 RECORDS
001700 001700     LABEL RECORDS ARE STANDARD.

```

Actions

Action	Description
File	Displays the File pull-down that is used to save and cancel changes, and to exit SmartEdit. The File pull-down is also used to access the Display and Update functions.
View	Displays the View pull-down that is used to access the Tree View facility and return to the Editor screen. View is also used to access the View - Paragraph Cross Reference pop-up and various Zoom functions.
Search	Displays the Search pull-down that is used to find, highlight, scroll, print, punch, and exclude a specified target.
Check	Displays the Check pull-down that is used to perform a syntax check on the programs and to set parameters for the Check facility. The Check facility is available only on the Editor screen.
List	Displays the List pull-down that is used to access list pop-ups that list information about calls, equates, performs, programs, or subsets. The List pull-down is available only on the Editor screen.

Action	Description
Options	Displays the Options pull-down that is used to customize your SmartEdit environment by setting certain parameters and options. (See "Options Pull-down" on page 156 for more information.)
Help	Displays the Help pull-down that is used to access the Online Help facility.

Pull-downs

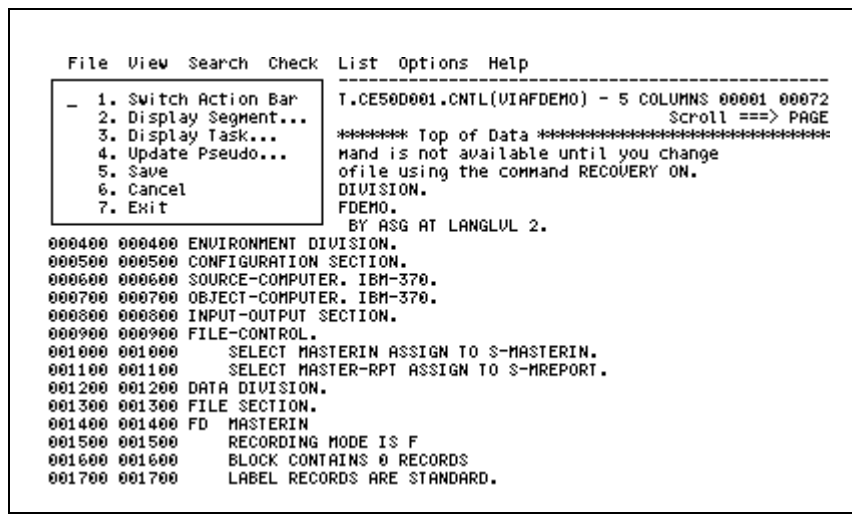
A pull-down, as shown in [Figure 7](#), displays when you select an action on the action bar. These are the two types of options on pull-downs:

- An option followed by an ellipsis (...) displays a pop-up when selected.
- An option not followed by an ellipsis immediately performs the requested function and usually redisplay the previous screen.

You can select an item on a pull-down using one of these methods:

- Move the cursor to the desired keyword and press Enter.
- Type the number of the desired action in the input field and press Enter.

Figure 7 • Pull-down Format (File Pull-down)



On the File pull-down, Display Segment, Display Task, and Update Pseudo display a pop-up when selected. Save, Cancel, and Exit do not display a pop-up, but immediately process the associated internal command.

Pop-ups

A pop-up is a window that displays as the result of selecting an item on a pull-down or another pop-up, or as the result of entering certain commands. A pop-up is superimposed over a screen, has a border, and usually does not have an action bar. Each pop-up contains one or more fields.

Follow the instructions on the pop-up to process the information. Use the END command (default PF3/15) to exit from a pop-up without processing it.

Figure 8 • Pop-up Format (Search Dataname Pop-up)

```

                                Search - Data Name

Type a data name and select search options. Then press Enter. For a
selection list, enter a pattern (e.g. ABC*) in the name area.

Data Name  _____

Data References      Indirect Impact      Size change
1  1. All           1  1. None             levels . . . ____
   2. Defs          2.  Of Size Change
   3. Uses           3.  Of Value Change
   4. Mods

Direction           Options                Action
1  1. All            - No Data Aliasing      1  1. Find
   2. Next           - IN-Clause...          2. Highlight
   3. Previous                          3. Scroll
   4. First                                                  4. Print
   5. Last                                                  5. Punch
                                                  6. Exclude

```

Pop-ups can contain Text and Selection fields. The field input area is designated by an underline. When a pop-up is first displayed, each field contains its default value, if applicable. On some pop-ups, if the default is changed the new value is retained for subsequent sessions. These are the types of fields:

Operand	Description
Text entry fields	Requires textual information, such as a dataname, program name, etc. The length of each text entry field is designated by an underline. In Figure 8 , Data Name is an example of a text entry field.
Selection fields	<p>Allows selection of one item from a list. Type the number of the desired selection in the field input area to the left of the first numbered item. In Figure 8, Direction is an example of a numbered selection field.</p> <p>Unnumbered selection fields are selected by entering a slash (/) in the field input area, and unselected by entering a blank space to remove the slash. No Data Aliasing is an example of an unnumbered selection field.</p>

Screens

A screen is a full-width display of information containing an action bar as the first line. SmartEdit screens are modeled after the ISPF/PDF edit screen, with the added features of CUA. See the next section for specific information about screen formats.

Editor Screen

When a program is opened in SmartEdit, an Editor screen, similar to [Figure 9](#), displays.

Figure 9 • SmartEdit Editor Screen

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==> Scroll ==> PAGE
***** Top of Data *****
-Warning- The UNDO command is not available until you change
your edit profile using the command RECOVERY ON.
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. VIAFDEMO.
000300 000300 AUTHOR. WRITTEN BY ASG AT LANGLVL 2.
000400 000400 ENVIRONMENT DIVISION.
000500 000500 CONFIGURATION SECTION.
000600 000600 SOURCE-COMPUTER. IBM-370.
000700 000700 OBJECT-COMPUTER. IBM-370.
000800 000800 INPUT-OUTPUT SECTION.
000900 000900 FILE-CONTROL.
001000 001000 SELECT MASTERIN ASSIGN TO $-MASTERIN.
001100 001100 SELECT MASTER-RPT ASSIGN TO $-MREPORT.
001200 001200 DATA DIVISION.
001300 001300 FILE SECTION.
001400 001400 FD MASTERIN
001500 001500 RECORDING MODE IS F
001600 001600 BLOCK CONTAINS 0 RECORDS
001700 001700 LABEL RECORDS ARE STANDARD.

```

ISPF 4.1 through 4.8

If you are using ISPF 4.1, the format of the Editor screen is controlled by ISPF settings. Use the ISPF command SETTINGS to display the ISPF Settings screen. After making the desired modifications, press PF3 to return to the Editor screen.

This table describes the features of a screen:

Area	Description
Action bar	Refers to the actions (File, View, Search, etc.) positioned at the top of every screen. See "Using the Action Bar" on page 6 for more information.
SmartEdit release number	Specifies the SmartEdit identification and release number.
COBOL program name	Specifies the fully-qualified name of the active COBOL program.

Area	Description
COLUMNS	Contains standard ISPF column information. This area of the screen is also used to display short informational and error messages.
COMMAND ===>	Identifies the command input area. Enter SmartEdit primary commands here.
SCROLL	<p>Specifies the number of screen lines or columns to scroll. This field is omitted from screens that cannot be scrolled. These are the valid values:</p> <ul style="list-style-type: none"> • 1-9999 A number from 1 to 9999 that specifies the number of lines or columns to scroll. • CSR Specifies a scroll value of one page from the current cursor location. • HALF Specifies a scroll value of a half page. • MAX Specifies that the top, bottom, right, or left margin is the scroll value. • PAGE Specifies a scroll value of one page. • DATA Specifies a scroll value of one line less than a page.
Long message field	Displays descriptive error and information messages. A long message displays if there is no corresponding short message, or if you enter the HELP command when a short message displays.

Concepts

SmartEdit extends the ISPF/PDF editor to accept and process an advanced set of commands by providing convenient facilities not available in the ISPF/PDF editor.

Screen Messages

SmartEdit messages are displayed on the line below the command input area. Short messages are displayed when available. Long messages are displayed if a short message does not exist, or when help is requested immediately following a displayed short message. Enter the HELP command followed by the message number to display an explanation and any actions for the message.

See ["Help" on page 175](#) for additional information about the Help facility.

Sets

A set is a grouping of source lines in a COBOL source program. Sets consist of subsets and line range sets (see ["Targets" on page 15](#) for information about line range sets). Many SmartEdit commands and actions make use of one or more of these sets or subsets. In addition, you can concatenate sets by placing a plus sign (+) between them. For example, to highlight the set of lines containing IO statements and PARAGRAPH labels, type this command:

```
HIGH IO + PAR
```

Subsets

There are three types of subsets: COBOL language subsets, screen subsets, and tagged lines subsets. Use the LIST SUBSETS command to display the List - COBOL Subset Names pop-up, which describes each subset.

COBOL Subsets

SmartEdit classifies COBOL statements into subsets by grouping together COBOL verbs of a similar nature. For example, you could refer to any lines that contain READ, WRITE, OPEN, or CLOSE verbs by using the COBOL language subset name IO.

For a complete list of the verbs included in each of the COBOL subset names, type LIST SUBSETS on the Editor screen to see the List - COBOL Subset Names pop-up ([Figure 112 on page 153](#)). This table describes each of the COBOL subsets:

COBOL Subset	Description
ASsignment	Includes statements that assign a value to a data item, such as MOVE, ADD, or COMPUTE.
CAll	Includes statements related to subprogram calls, such as CALL and CANCEL.
CIcs	Specifies any CICS (Customer Information Control System) or DL/I command level statements.
COBOLII	Includes PROCEDURE DIVISION statements that are exclusively COBOL II, including CONTINUE, END, and INITIALIZE verbs.
COBOL/370	Specifies statements and clauses unique to COBOL/370, such as intrinsic function calls, procedure pointers, and calls to LE/370 run-time environment.
COMment	Specifies statements having no run-time effect such as all lines with an asterisk (*) in column 7, the entire IDENTIFICATION DIVISION and NOTE statements.

COBOL Subset	Description
CONditional	Includes statements or parts of statements that conditionally change the flow of control in a program such as IF, ELSE, and WHEN.
DB2 SQL	Specifies all Exec SQL statements.
DDL	Includes data definition language statements, such as CREATE, ALTER, DECLARE, and DROP
DEBug	Includes statements containing a DEBUG, EXHIBIT, ON, READY, or RESET verb, as well as statements containing a D in column 7.
DEFinition	Specifies declaratives of data items, including the SPECIAL-NAMES paragraph in the ENVIRONMENT DIVISION, the entire DATA DIVISION, and statements related to defining DB2 items (i.e., CREATE, ALTER, and DECLARE).
DIRective	Includes statements that direct the compiler to take specific actions during compilation such as BASIS, EJECT, and TITLE.
DL/I	Includes EXEC DL/I statements, such as ENTRY 'DLITCBL' and CALL 'CBLTDLI'.
DML	Specifies data manipulation language statements, such as SELECT, UPDATE, INSERT, and COMMENT.
ENtry	Specifies the PROCEDURE DIVISION statement and all ENTRY statements.
EXIt	Specifies statements containing a STOP RUN, GOBACK, or EXIT PROGRAM verb, as well as CALL statements that are indicated as NORET (non-returning).
GOto	Includes statements containing an ALTER or GOTO verb.
IDMS	Includes all CA-IDMS statements.
IMS	Includes exec DL/I statements, such as CALL 'CBLTDLI' and ENTRY 'DLITCBL'.
IO Input Output	Specifies COBOL IO statements (IO, Input, or Output respectively) including CALL statements that are indicated as containing IO, Input, or Output.

COBOL Subset	Description
LABel DIVision PARagraph SEction	Includes statements containing DIVISION or SECTION headers, or PARAGRAPH labels. LABEL refers to the PROCEDURE DIVISION line and all section and paragraph names in the PROCEDURE DIVISION.
MATH	Specifies statements containing ADD, SUBTRACT, MULTIPLY, DIVIDE, or COMPUTE verbs.
PERform	Includes statements containing the PERFORM, SORT, or MERGE verbs.
SORTMerge	Includes statements containing SORT, MERGE, or RELEASE verbs; paragraph or section names referred to in INPUT/OUTPUT PROCEDURES.
STructure	Specifies a group of COBOL subsets that help show the general structure of the program. These COBOL subsets include: CALL, PERFORM, DIVISION, SECTION, PARAGRAPH, EXIT, and GOTO.

Screen Subsets

Screen subsets are generally the result of an interactive command. To specify one of these subsets, type the entire name, or at least the minimum abbreviation as indicated by the upper case letters:

Highlighted | HI
 NONHighlighted | NHI
 Excluded | X
 NONExcluded | NX

Tag Subsets

Tag subsets are displayed in columns 1 through 6 of the Editor screen to provide you with immediate information about the source code. The tags displayed in columns 1 through 6 are the results of commands. You can use tags as subsets in commands that accept subsets as targets.

Targets

A target is the object of a SmartEdit primary command. Targets are defined in these categories:

- Program name
- Subset name
- Line range
- Label name
- Dataname
- Pattern string

Subset Name

A subset name is one of the previously described COBOL language subsets, screen subsets, or tag subsets.

Line Range

A line range is a single line or a group of lines. Line ranges are specified by placing a hyphen (-) between the first and last line numbers in the range (e.g., 214 - 376).

Line numbers are shown in the first six columns of the screen. If the specified line number is greater than the last line in the program, the last line is assumed.

You can use the ISPF label function in place of line numbers for any command that accepts the line range operand.

Label Name

A label name is any paragraph or section name of the PROCEDURE DIVISION, as well as the literals PROCEDURE and PROC.

Dataname

A dataname can be any of these items:

- Elementary dataname
- File name
- Group name
- Table name
- Table element name
- Special name

You can specify any legal COBOL reference for a data element as a dataname. If a variable is redefined to another name, SmartEdit searches for the specified variable name and the redefined name. Any reference to an entry in a table is treated as a reference to the entire table. When data items overlap so a name can refer to parts of multiple data items, searches are performed on each part and all references are reported. For example, if a group item is specified in a search command, references to the group item as well as the individual elements within the group are located. This is also true of modifications, uses, or references to the data item. SmartEdit locates valid references to the variable item as opposed to simple pattern matching of the characters in the variable name. These references are called aliases.

Specify fully-qualified datanames by following them with a standard COBOL OF clause, followed by the group level dataname. For example:

```
DATA-NAME-ELEMENT OF DATA-NAME-GROUP
```

or for COBOL II Release 3 or later programs:

```
DATA-NAME ELEMENT OF SUBPROGRAM  
DATA-NAME ELEMENT IN SUBPROGRAM
```

Locate multiple datanames at the same time by concatenating the datanames with a plus sign (+) between them. For example:

```
DATA-NAME1 + DATA-NAME2
```

Specify datanames using one of these subordinate operands:

Operand	Description
MOD	Specifies occurrences of a data item where its value is being set or altered.
USE	Specifies occurrences of a data item where its value is being tested or used.
DEF	Specifies definitions of a data item and its aliases as specified in the DATA DIVISION.
REF	Identifies all MODIFICATION and USE occurrences. REFERENCE also includes DEFINITION occurrences on some commands. This is the default usage for datanames.

The EXCLUDE, FINDXTND, HIGH, LPRINT, LPUNCH, and SCROLL commands also offer ALIAS/NOALIAS and DIRECT/INDIRECT operands. These operands are used as shown in this table:

Operand	Description
ALIAS/NOALIAS	ALIAS includes all aliases for the specified dataname and is the default.
DIRECT/INDIRECT	<p>DIRECT includes only the specified dataname. INDIRECT locates any dataname indirectly affected by the specified dataname (and aliases if specified). Further qualify INDIRECT data items using SIZE, VALUE, and LEVELS subordinate operands.</p> <ul style="list-style-type: none"> • SIZE. Occurrences of a dataname indirectly affected by a change in the size of the dataname. SIZE is the default operand for INDIRECT. • VALUE. Occurrences of a dataname directly or indirectly affected by a change in the value of the dataname. • LEVELS. Identifies the depth of the direct references. If LEVELS is not specified, the default is all levels.

Pattern String

A pattern string is a sequence of characters, surrounded by single or double quotes if it contains blanks. Specify strings of non-alphanumeric characters using the X'string', T'string', and P'string' operands. Further qualify the string using the WORD, PREFIX, or SUFFIX subordinate operands.

Operand	Description
X'string'	Specifies a hexadecimal string, enclosed in single or double quotes.
T'string'	Specifies a text string, which disregards upper and lower case, enclosed in single or double quotes.

Operand	Description
P' <i>string</i> '	Specifies a picture string, enclosed in single or double quotes. These are the valid picture strings: P'=' Any character P'-' Any nonblank character P'.' Any nondisplay character P'#' Any numeric character P'-' Any non-numeric character P'@' Any alphabetic character (upper or lowercase) P'<' Any lowercase alphabetic character P'>' Any uppercase alphabetic character P'\$' Any special character (not alphabetic or numeric)
WORD	Specifies a string preceded and followed by any non-alphanumeric character (except a hyphen).
PREFIX	Indicates a word that begins with the specified string.
SUFFIX	Indicates a word that ends with the specified string.

PF Key Values

SmartEdit uses your existing ISPF PF keys. ASG recommends that you reassign a few of your default ISPF keys to make optimum use of SmartEdit.

To edit the PF keys

- 1 Type KEYS to display the ISPF PF Key Definitions and Labels screen.
- 2 Makes these changes to the default PF key values:

PF Key	Action
PF01/13	HELP
PF02/14	SPLIT
PF03/15	END
PF04/16	RTREEVW
PF05/17	RFIND
PF06/18	RCHANGE

PF Key	Action
PF07/19	UP
PF08/20	DOWN
PF09/21	SWAP
PF10/22	BRANCH
PF11/23	BRANCH BACKUP
PF12/24	RECALL

ASG suggests these alternative values for the PF keys:

PF KEY	ACTION
PF01/13	ZOOMIN
PF04/16	ZOOM DEF
PF12/24	ZOOMOUT

3

Getting Started

This chapter describes how to begin using SmartEdit and contains these sections:

Section	Page
Introduction	22
Activating SmartEdit	22
Defining Options - COPY/Include Libraries	24
Defining Options - Product Parameters	31
Using Edit Entry Screens	32
Editor Screen	34
Tracing Program Control Flow	35
Viewing Copybooks In-context	40
Checking Program Syntax	43
Using the COBOL Intelligent Search Feature	46
COBOL Program Understanding Features	53
Defining Character Substitution Name	72
Troubleshooting	73

Introduction

The steps necessary to invoke SmartEdit vary by site. Check with the SmartEdit system coordinator at your site for these details. Before beginning the sample SmartEdit session presented in this section, you might find it helpful to review SmartEdit terminology listed in the ["Glossary" on page 317](#).

Throughout this section, the VIAFDEMO COBOL program is used to demonstrate the SmartEdit commands and actions. Depending on how your system is configured, typically, you can find VIAFDEMO in ASG.VIACEN`xx`.CNTL (where `xx` is the current Center release level).

The first time you activate SmartEdit you must define some of its parameters. You define these user options using a series of pull-downs and pop-ups to customize the SmartEdit environment. These are the tasks for customizing the SmartEdit environment:

- Activate a SmartEdit session using the COBEDIT command. For details, see ["Activating SmartEdit" on page 23](#).
- Define the source library manager (i.e., partitioned dataset, Librarian, Panvalet), COBOL language version, and COPY/INCLUDE dataset name(s) and source type in the Options - COPY/Include Libraries screen. For details, see ["Defining Options - COPY/Include Libraries" on page 24](#)
- Set and verify the compiler and preprocessor load member names and load libraries in the Options - Product Parameters pop-up. For details, see ["Options" on page 155](#).
- Set and verify the Log, List, Punch, and Work file allocations on the Options - Product Allocations pop-up. For details, see ["Options" on page 155](#).
- Set and verify the Log, List, and Punch file processing options and provide a JOB card on the Options - Log/List/Punch Definition pop-up. For details, see ["Options" on page 155](#).
- Set desired processing modes such as LEARN, XMODE, IDMS, or DB2 on the Options - Processing Modes pop-up. For details, see ["Options" on page 155](#).
- Set and verify COBOL syntax check parameters using the ["Check Pull-down" on page 137](#).

Activating SmartEdit

The method that you use to activate SmartEdit depends on your system setup. Contact your Systems Administrator if you have any questions.

To activate SmartEdit at most sites, follow this step:

- Type the ISPF/PDF editor as you normally would, or make the appropriate selection from the ISPF primary menu.

Typically, SmartEdit is automatically activated depending on the profile of the dataset being edited. The SmartEdit identification displays in the upper left corner of the display to indicate that SmartEdit is active.

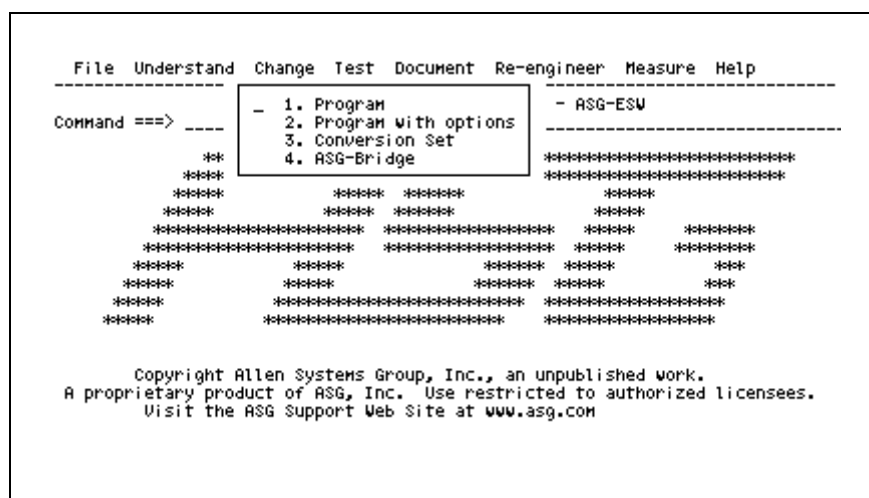
If SmartEdit does not automatically activate, follow this step:

- Type COBEDIT in the command input area on the ISPF/PDF or source manager editor.

If your site uses the ESW screen to start SmartEdit, select Change ► Program to display the Edit - Entry screen, or Change ► Program with Options to display the Options - COPY/Include Libraries screen.

[Figure 10](#) shows the ESW selections for SmartEdit. If you start SmartEdit from the ESW Primary screen, the product name displays on the screens as ASG-ESW - COBOL Editing.

Figure 10 • Existing Systems Workbench Primary Screen



SmartEdit allows you to edit COBOL programs stored as a partitioned dataset or using source managers such as Librarian and Panvalet. The source manager is specified on the Options - COPY/Include Libraries screen. Enter the editor selection for the source manager if SmartEdit has been installed to be globally available. See ["Defining Options - COPY/Include Libraries" on page 24](#) for more information.

Defining Options - COPY/Include Libraries

Use the Options - COPY/Include Libraries screen to specify options to be used for the editor. These options are included:

- The source manager to be used, such as PDS, Panvalet, and Librarian
- The COBOL language version
- The editor parameters
- The COPY/INCLUDE dataset name(s) and source manager type

To display the Options - COPY/Include Libraries screen at any time during the session

- 1 Select Options ► Refresh.

Or

Type REFRESH on the editor screen.

The Options - COPY/Include Libraries screen, shown in [Figure 11](#), displays.

Note:

Depending on the method of activating SmartEdit in your installation, this screen may also be displayed on entry into SmartEdit.

Figure 11 • Options - COPY/Include Libraries Screen

```

File COBOL SourceManager Options
-----
Options - COPY/Include Libraries
Command ==> _____ Scroll ==> 3
Specify Source Manager, COBOL Version and COPY/Include Libraries.
Source Manager . . : PDS/Sequential   COBOL Version . : COBOL II R3

COPY List:
Data Set Name . . :
Member . . . . . :
Description . . . :

Copy/Include Data Set Name(s)      Type VolSer   Unit   Password
-----
/ 'USER.TEST.CNTL'                PDS
***** BOTTOM OF DATA *****

```

- 2 Specify the correct options for your source manager and complete the necessary information in the Copy/Include Dataset Name(s) fields. Selecting File ► Save COPY List lets you save and retrieve lists of COPY/Include libraries for use in different edit sessions.

- 3 Press Enter to return to the editor screen. A message in the short message area indicates REFRESH COMPLETED.

Actions

Use the COPY/Include Libraries screen action bar to manage COPY lists, change option settings for the COPY/Include libraries, and modify SmartEdit product settings. These are the actions available on the action bar:

Actions	Description
File	<p>Enter the number of the desired action:</p> <ol style="list-style-type: none"> 1. Load COPY List. Displays the File - Load COPY list pop-up to load a previously defined list of COPY/Include libraries. (See "Loading the COPY List" on page 29 for more information.) 2. Save COPY List. Displays the File - Save COPY List pop-up to save the list of libraries specified on the Options - COPY/Include Libraries screen. (See "Saving the COPY List" on page 29 for more information.) 3. Extract COPY List. Displays the File - Extract COPY List pop-up to extract COPY/Include library information from compile JCL. (See "Extracting a COPY" on page 30 for more information.) 4. Exit. If a SmartEdit program edit session is active, selecting Exit displays the editor screen. If an edit session is not active, selecting exit closes SmartEdit.
COBOL	<p>Type the number of the appropriate version of COBOL:</p> <ol style="list-style-type: none"> 1. COBOL68 2. COBOL74 3. COBOL II 4. COBOL II R3 5. COBOL II R4 6. COBOL/370 7. COBOL MVSVM 8. COBOL OS/390 9. Enterprise COBOL

Actions	Description
Source Manager	Type the list number corresponding to the source manager that contains your program: <ol style="list-style-type: none"> 1. PDS/Sequential Dataset 2. Librarian 3. Panvalet 4. SCLM 5. Other (user-defined)
Options	Type the number of the desired action: <ol style="list-style-type: none"> 1. Product Parameters. Displays the Options - Product Parameters pop-up to set parameters to access DB2, IDMS, or DATACOM screens from SmartEdit. (See "Options - Product Parameters Pop-up" on page 157 for more information.) 2. Product Allocations. Displays the Options - Product Allocations pop-up to specify DASD volumes for the Log, List, Punch, and Work files, and to specify space for the Work file. (See "Options - Product Allocations Pop-up" on page 164 for more information.) 3. Processing Modes. Displays the Options - Processing Modes pop-up to enable and disable SmartEdit processing modes (LEARN, XMODE, DB2, and IDMS). (See "Options - Processing Modes Pop-up" on page 169 for more information.) 4. CHECK Parameters. Displays the Check - Parameters pop-up to set compiler options, specify preprocessors, and define load modules and load libraries for the edit session. 5. ASG-SmartEdit Parm. Displays the Options - ASG-SmartEdit Parameters pop-up to override editor parameters set during SmartEdit installation. (See "Defining Options - Product Parameters" on page 31 for more information.)

Fields

Field	Description
Source Manager	Specifies the source manager that contains your program. Select Source Manager from the action bar to change the source manager.
COBOL Version	Specifies the COBOL version for your program. Select COBOL from the action bar to change the COBOL version.

Field	Description																
Dataset Name	Displays the name of the dataset containing the copy list member if you specified a copy list. To load a copy list, select File ► Load COPY List. To import a copy list from compile JCL, select File ► Extract COPY List.																
Member	Displays the name of the copy list member if you specified a copy list.																
Description	Displays the description of the specified copy list, if available.																
Copy/Include Dataset Name(s)	<p>Type the COPY/Include dataset information as necessary for your program. To save the dataset information in a copy list that can be retrieved in another edit session, select File ► Save COPY List.</p> <p>To load COPY/Include dataset information from an existing copy list, select File ► Load COPY List.</p> <p>To import COPY/Include library information from compile JCL, select File ► Extract COPY List.</p>																
Line command area	<p>Specifies commands you can use to edit the dataset names. Enter these commands in the field to the left of the Copy/Include Dataset Name(s) field. These are the valid commands:</p> <table> <tr><td>C</td><td>Copy</td></tr> <tr><td>M</td><td>Move</td></tr> <tr><td>I</td><td>Insert</td></tr> <tr><td>D</td><td>Delete</td></tr> <tr><td>A</td><td>After</td></tr> <tr><td>R</td><td>Repeat</td></tr> <tr><td>B</td><td>Before</td></tr> <tr><td>/</td><td>Enable COPY library</td></tr> </table>	C	Copy	M	Move	I	Insert	D	Delete	A	After	R	Repeat	B	Before	/	Enable COPY library
C	Copy																
M	Move																
I	Insert																
D	Delete																
A	After																
R	Repeat																
B	Before																
/	Enable COPY library																
Type	<p>Specifies the COPY or INCLUDE dataset source type. These are the valid entries:</p> <table> <tr><td>PDS</td><td>Partitioned dataset or sequential dataset</td></tr> <tr><td>LIB</td><td>Librarian</td></tr> <tr><td>PAN</td><td>Panvalet</td></tr> <tr><td>USR</td><td>User-defined</td></tr> </table>	PDS	Partitioned dataset or sequential dataset	LIB	Librarian	PAN	Panvalet	USR	User-defined								
PDS	Partitioned dataset or sequential dataset																
LIB	Librarian																
PAN	Panvalet																
USR	User-defined																
VolSer	Specifies the volume serial number containing the COPY or INCLUDE dataset if the dataset is not cataloged.																

Field	Description
Unit	Specifies the generic storage unit used by the COPY or INCLUDE dataset.
Password	Specifies the password for the COPY or INCLUDE dataset if applicable.

Usage Notes

Once you enter information on the Options - COPY/Include Libraries screen, it is saved and displayed the next time the screen is presented.

Note:

When editing a Panvalet or Librarian member, the ++INCLUDE or -INC expansion is set to NO. This allows SmartEdit to expand copy members when necessary.

When you specify a dataset name, it is automatically enabled. To disable a copy library for the current edit session, type a blank over the / in the line command area.

If you are using a COPY list and make changes to the dataset information on the COPY/Include Libraries screen, the changes are only active in the current edit session. To save the changes permanently in the COPY list member, select File ► Save COPY list.

If the Options - COPY/Include Libraries screen displays automatically on entry to SmartEdit, press Enter to display the appropriate Edit Entry screen for your Source Manager.

Create COPY List Syntax - COPY lists online by selecting File ► Save COPY List, or by using an external editor. COPY lists created using an external editor must follow this COPY list syntax (text enclosed in brackets is optional):

```
ASG-SMARTEDIT COPYLIST (mm/dd/yyyy by userid)
DESC='xxxxxxxxxx...xxxxxxxxx'
DSN='dataset #1' TYPE=xxx {V=xxxxxxxx} {U=xxxx} {P=xxx}
DSN='dataset #2' TYPE=xxx {V=xxxxxxxx} {U=xxxx} {P=xxx}
ASG-SMARTEDIT ENDCOPYLIST
```

Loading the COPY List

To load a previously defined list of COPY/Include libraries, follow this step:

- ▶ Select File ▶ Load COPY List. The File - Load COPY List pop-up, shown in [Figure 12](#), displays.

Figure 12 • File - Load COPY List Pop-up

```

File - Load COPY List
Type dataset name and member of COPY List to load.
Data Set Name . . . _____
  
```

Fields

Field	Description
Data Set Name	Type the dataset name and member of the COPY list to be loaded.

Saving the COPY List

To save the list of libraries specified on the Options - COPY/Include Libraries screen, follow this step:

- ▶ Select File ▶ Save COPY List. The File - Save COPY List pop-up, shown in [Figure 13](#), displays.

Figure 13 • File - Save COPY List Pop-up

```

File - Save COPY List
Type dataset name and member for COPY List.
Data Set Name . . . _____
Description . . . _____
Options:
_ COBOL Comment Format
  
```

Fields

Field	Description
Data Set Name	Type the PDS dataset and member name where the COPY list is to be stored.
Description	Type a description for the COPY list.

Options

Option	Description
COBOL Comment Format	Select this option to format the COPY list as a COBOL comment.

Extracting a COPY

To extract a list of COPY/Include libraries from the compile JCL, follow this step:

- Select File ► Extract Copy List. The File - Extract COPY List pop-up, shown in [Figure 14](#), displays.

Figure 14 • File - Extract COPY List Pop-up

File - Extract COPY List

Type Compile JCL dataset name and Member for COPY List.

Data Set Name . . . _____

Fields

Field	Description
Data Set Name	Type the compile JCL dataset name and member for the COPY list.

Usage Notes

The Extract Copy List option extracts the names of the SYSLIB datasets from the specified JCL for the compiler step ONLY. The extract process submits the specified JCL with a TYPRUN=SCAN, and parses the output to determine the dataset names used in SYSLIB for the compile step. The dataset names are then placed on the COPY/Include Libraries dataset name screen.

Note:

The JCL must be valid and must execute IGYCRCTL.

Defining Options - Product Parameters

To override the editor parameters specified at product installation, follow this step:

- Select Options ► Product Parameters. The Options - Product Parameters pop-up, shown in [Figure 15](#), displays.

Figure 15 • Options - Product Parameters Pop-up

```

Options - Product Parameters

Alarm . . . . . YES (Yes or No)
Cursor Character % (Token sub. character)

Data Base Parameters
-- 1. DB2
   2. IDMS
   3. DATACOM
  
```

Fields

Field	Description
Parameters	Type the parameters to be overridden using the options listed below. To concatenate parameters, separate them with a comma or a space.
Input(x,x,...x) Input=x NOInput(x,x,...x) NOInput=x	<p>INPUT allows you to list the names of CALLED programs that contain Input statements. When commands that search for Input are issued, the statements that call these programs are shown in the results of the command. The programs specified are in addition to those listed at installation time.</p> <p>NOINPUT allows you to override the installation default and delete names from the list of CALLED programs that contain Input statements.</p>

Field	Description
IO(x,x,...x) IO=x NOIO(x,x,...x) NOIO=x	IO allows you to list names of CALLED programs that contain Input and/or Output statements. When Input and/or Output search commands are issued, the statements calling the specified programs display in the search results. Programs specified are in addition to those listed at installation time. NOIO allows you to override the installation default and delete names from the list of CALLED programs that contain Input and/or Output statements.
LIBCopy NOLIBCopy	Allows COPY members from Librarian datasets to be expanded. The default is LIBCOPY.
Output(x,x,...x) Output=x NOOutput(x,x,...x) NOOutput=x	OUTPUT allows you to list the names of CALLED programs that contain Output statements. When commands that search for Output are issued, the statements that call these programs are shown in the results of the command. The programs specified are in addition to those listed at installation time. NOOUTPUT allows you to override the installation default and delete names from the list of CALLED programs that contain Output statements.
PANCopy NOPANCopy	Allows COPY members from Panvalet datasets to be expanded. If your Panvalet source uses ++INCLUDE, specify NOPANCOPY. If your Panvalet source uses COPY, specify PANCOPY. The default is NOPANCOPY.
RETurn(x,x,...x) RETurn=x NORETurn(x,x,...x) NORETurn=x	RETURN allows you to change names of the programs or entry points that do not return when CALLED. It overrides the system defaults and allows you to list the names of the programs or entry points that you want to return. NORETURN allows you to list names of programs or entry points that do not return. When any of the listed programs are called by the program being edited, SmartEdit treats them as non-returning calls. The programs specified are in addition to the system defaults for programs that do not return.

Using Edit Entry Screens

This section describes entry screens for ISPF/PDS, Panvalet, and Librarian. They are only samples and may not reflect the current releases for your site. SmartEdit does not control these screens. See your site documentation or check with your SmartEdit system coordinator for current information about the source managers that are available for your use.

ISPF/PDS Edit - Entry Screen

The Edit Entry screen for ISPF is shown in [Figure 16](#).

Figure 16 • Edit - Entry Screen

```

Menu RefList RefMode Utilities LMF Workstation Help
-----
Edit Entry Panel
Command ==> _____

ISPF Library:
Project . . . UIAINMM
Group . . . CE70TEST
Type . . . CNTL
Member . . . _____ (Blank or pattern for member selection list)

Other Partitioned, Sequential or VSAM Data Set:
Data Set Name . . . _____
Volume Serial . . . _____ (If not cataloged)

Workstation File:
File Name . . . _____

Initial Macro . . . _____ Options
Profile Name . . . _____ - Confirm Cancel/Move/Replace
Format Name . . . _____ - Mixed Mode
Data Set Password . . . _____ - Edit on Workstation
                                     - Preserve VB record length

```

Complete the ISPF library information and type the password and profile name, if required.

Panvalet Edit - Entry Screen

The Edit Entry screen for Panvalet is shown in [Figure 17](#).

Figure 17 • Panvalet Entry Screen

```

----- PVEDIT - ENTRY PANEL ----- Computer Associates
COMMAND ==>
                                     VERSION - 1439811

Standard CA-Panvalet Library:
PROJECT ==>
GROUP ==>
TYPE ==>

MEMBER ==> _ (Blank for Criteria Selection PANEL for MSLs)
              (Enter name with Wildcard * ? notation for MSL)

Non-Standard CA-Panvalet Library - RETAIN ==> (Y/N, Save in Profile)
DSNAME ==>
VOLSER ==> (If NOT Cataloged)

CA-Panvalet Retrieval/Save Options:
CONTROL ==> (If necessary) EXPAND ==> (Y/N)
ACCESS ==> (If necessary) EDIT LOCK ==> (Y/N)
SAVE ==> (1 = First Library
          0 = Origin Library)

CA-Panvalet EDIT Profile ==> Initial Macro ==>

Press ENTER key to process; Enter END command to terminate.

```

If the program name is not entered on the Panvalet Edit - Entry screen, the member selection list displays.

Librarian Entry Screen

The Edit Entry screen for Librarian is shown in [Figure 18](#).

Figure 18 • Librarian Entry Screen

```

***** 4.2 EXTENDED LIBRARIAN INTERACTIVE PRODUCTIVITY SERVICES *****
COMMAND ==>

ELIPS *****AVAILABLE FUNCTIONS*****
* A- ARCLIST  E- EDIT      *
* B- BROUSE   I- INFO     *
* C- COPY     J- JCLVAL    *
* D- DELETE   P- PRINT     *
*              R- RENAME   *
*              *            *
* JCLOPTS- JCLVAL OPTIONS *
* OPTIONS- DISPLAY/SET   *
* TYPES - AVAILABLE TYPES *
* UTILITY- MOVE/COPY     *
***** SCREEN 1 *****

LIBRARIAN LIBRARY:
PROJECT ==>
GROUP ==>
TYPE ==>
MEMBER ==>          NEWNAME ==>
PSUD ==>            MCD ==>
ARC LVL ==>

OTHER LIBRARIAN DSNAME ==>

SELECTION LIST ==> (L/S/Q)  LONG LIST:  PGMR ==>
                                         TYPE ==> (OR /lan)
CONFIRM DELETE ==> (Y/N)    PRINT FUNCTION ==> (REPT/DATA)
DISPLAY HISTORY ==> (Y/N)   INITIAL EDIT MACRO ==>

```

Complete the fields in the Librarian LIBRARY section with correct dataset and member information.

Editor Screen

After you press Enter on the appropriate Entry screen, the editor screen, shown in [Figure 19](#) displays for the program you selected.

Figure 19 • Editor Screen

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==> _____ $scroll ==> PAGE
***** Top of Data *****
-Warning- The UNDO command is not available until you change
your edit profile using the command RECOVERY ON.
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. VIAFDEMO.
000300 000300 AUTHOR. WRITTEN BY ASG AT LANGLVL 2.
000400 000400 ENVIRONMENT DIVISION.
000500 000500 CONFIGURATION SECTION.
000600 000600 SOURCE-COMPUTER. IBM-370.
000700 000700 OBJECT-COMPUTER. IBM-370.
000800 000800 INPUT-OUTPUT SECTION.
000900 000900 FILE-CONTROL.
001000 001000 SELECT MASTERIN ASSIGN TO S-MASTERIN.
001100 001100 SELECT MASTER-RPT ASSIGN TO S-MREPORT.
001200 001200 DATA DIVISION.
001300 001300 FILE SECTION.
001400 001400 FD MASTERIN
001500 001500 RECORDING MODE IS F
001600 001600 BLOCK CONTAINS 0 RECORDS
001700 001700 LABEL RECORDS ARE STANDARD.

```

If the SmartEdit CUA action bar is not displayed on the top of the editor screen, type the CUA command in the command input area of the editor screen.

If you prefer to use commands for all SmartEdit functions, you can remove the CUA action bar by typing CUA OFF in the command area on the editor screen.

All examples shown in this section assume that the SmartEdit CUA action bar displays.

ISPF 4.1 through 4.8

If you are using ISPF 4.1 through 4.8, the format of the editor screen is controlled by ISPF settings. Use the ISPF command SETTINGS to display the ISPF Settings screen. After making the desired modifications, press PF3 to return to the editor screen.

You are now ready to use the editor.

Tracing Program Control Flow

The Branch function is very helpful when tracing branching logic. Use this function to trace branching logic several levels deep and then backup to each PERFORM, GO TO, or CALL statement previously selected. The Branch function is available on the Search pull-down or by using primary or line commands.

An effective technique for using the Branch function is to assign the BRANCH command to a PF key. The branch can then be accomplished by positioning the cursor on a PERFORM, GO TO, or CALL statement and pressing the PF key.

Use the Branch Backup function to return to the statement from which the branch occurred. This function is available on the Search pull-down from the CUA action bar or using the BRANCH BACKUP primary command.

If a PROGRAM EXIT or STOP RUN statement is encountered, a message displays indicating that a logical end has been reached.

Following the Flow of Program Logic

To locate the COBOL PROCEDURE DIVISION using CUA

- 1 Select Search ► Branch and press Enter. The Search - Branch Request pop-up, shown in [Figure 20](#), displays.

Figure 20 • Locating the PROCEDURE DIVISION Header

```

Search - Branch Request

Type the Target and select the Target Type. Then press
Enter. For a selection list, type a pattern in the
Target field.

Target  PROC_-----

Target Type
--  1. Cursor Position
    2. Label
    3. Program
    4. Backup
  
```

- 2 Type PROC in the Target field on the Search - Branch Request pop-up and press Enter.

You may also use the BRANCH command to locate the PROCEDURE DIVISION. To do this, type BRANCH with no operands and position the cursor in a part of the program other than the PROCEDURE DIVISION.

The display is positioned at the PROCEDURE DIVISION header, as shown in [Figure 21](#).

Figure 21 • Results of the Branch Option for PROC

```

File View Search Check List Options Help
ASG-SmartEdit - EDIT - VMAINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072)
Command ==>
020600 020600 EJECT
020700 020700 PROCEDURE DIVISION.
020800 020800
020900 020900*****
021000 021000* PERFORM PROGRAM INITIALIZATION *
021100 021100*****
021200 021200
021300 021300 PERFORM PROGRAM-INIT.
021400 021400
021500 021500*****
021600 021600* DRIVER SUBROUTINE LOOP *
021700 021700*****
021800 021800
021900 021900 PERFORM P000-NEXT THRU P000-EXIT
022000 022000 VARYING REC-CNT FROM 1 BY 1
022100 022100 UNTIL END-INPUT.
022200 022200
022300 022300 DISPLAY 'TOTAL INPUT RECORDS - ' REC-CNT.
022400 022400 DISPLAY 'END VIAFDEMO PROCESSING' UPON CONSOLE.
022500 022500
  
```

Following Program Flow

As you read through the program logic, you come to the perform of paragraphs P000-NEXT through P000-EXIT. Since it is unclear what these paragraphs do, you want to locate the source code for the first paragraph in the PERFORM THRU statement, P000-NEXT.

To locate the source code for this paragraph using CUA

- 1 Select Search ► Branch and press Enter.
- 2 Type the paragraph name P000-NEXT in the Target field on the Search - Branch Request pop-up, select Label as the Target Type to indicate that the target is a paragraph, and press Enter.

As shown in [Figure 22](#), the resulting display is positioned at the paragraph P000-NEXT.

Figure 22 • Locating a Paragraph Label

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==>
025400 025400
025500 P000-NEXT.
025600 025600
025700 025700 CALL 'VIAFDEM1' USING MASTER-IN
025800 025800 MASTER-END-OF-FILE
025900 025900 MASTER-REPORT-DATE.
026000 026000
026100 026100 PERFORM P100-PRINT
026200 026200 THRU P119-EXIT.
026300 026300
026400 026400 PERFORM P120-READ
026500 026500 THRU P129-EXIT.
026600 026600
026700 026700 P000-EXIT.
026800 026800 EXIT.
026900 026900 EJECT
027000 027000*****
027100 027100* VALIDATE THE INPUT PARM CODE *
027200 027200*****
027300 027300

```

You can accomplish this result using the command method, by typing BRANCH, placing the cursor on the line containing PERFORM P000-NEXT, and pressing Enter.

Within the paragraph P000-NEXT, there is a PERFORM of P100-PRINT. From the name, you expect that paragraph to do the printing for the program.

- 3 Repeat [step 1 on page 37](#) through [step 2 on page 37](#) using the CUA function to view the source code in P100-PRINT for analysis.

Or

Type BR on the command line containing the PERFORM P100-PRINT statement and press Enter.

As shown in [Figure 23](#), the resulting display is positioned at the paragraph P100-PRINT.

Figure 23 • Results of Locating a Label

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE500001.CNTL(VIA *Bottom of data reached*
Command ==> Scroll ==> PAGE
029900
030000 030000 P100-PRINT.
030100 030100
030200 030200* IF THE FIRST DIGITS OF THE ZIP CODE CHANGE, PRINT THE SUBTOTAL:
030300 030300* AND SKIP TO THE TOP OF THE PAGE.
030400 030400
030500 030500 P105-NOT-FIRST-TIME.
030600 030600
030700 030700 MOVE ZIP-CODE TO HLD-ZIP.
030800 030800 IF HLD-ZIP-PREFIX EQUAL CUR-PREFIX
030900 030900 NEXT SENTENCE
031000 031000 ELSE
031100 031100 PERFORM P150-SUBTOT
031200 031200 THRU P169-EXIT
031300 031300 MOVE HLD-ZIP-PREFIX TO CUR-PREFIX.
031400 031400
031500 031500 IF LINE-CNT GREATER THAN 53
031600 031600 PERFORM P160-HDG
031700 031700 THRU P169-EXIT.
031800 031800

```

- 4 Continue to follow the program flow by typing BR beside any statement containing a GO TO, PERFORM, or CALL.

Returning to Previous Branch Locations

After reviewing the code in P100-PRINT, return to the P000-NEXT paragraph to continue your review of the program logic.

To return to previous Branch locations

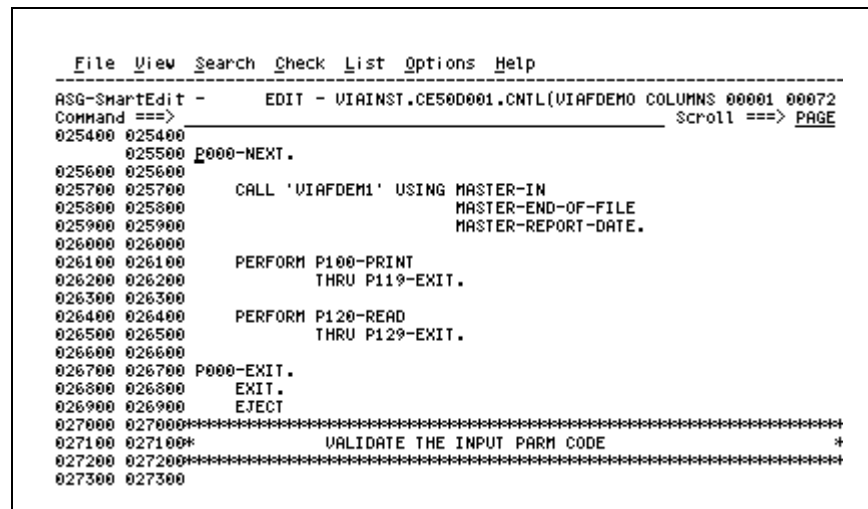
- 1 Select Search ► Branch to display the Search - Branch Request pop-up. Select Backup and press Enter.

Or

Type BRANCH BACKUP on the command line and press Enter.

As shown in [Figure 24](#), the display is positioned at the paragraph previously located, P000-NEXT.

Figure 24 • Returning to a Prior Paragraph Label



```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE500001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==> Scroll ==> PAGE
025400 025400
025500 P000-NEXT.
025600 025600
025700 025700 CALL 'VIAFDEM1' USING MASTER-IN
025800 025800 MASTER-END-OF-FILE
025900 025900 MASTER-REPORT-DATE.
026000 026000
026100 026100 PERFORM P100-PRINT
026200 026200 THRU P119-EXIT.
026300 026300
026400 026400 PERFORM P120-READ
026500 026500 THRU P129-EXIT.
026600 026600
026700 026700 P000-EXIT.
026800 026800 EXIT.
026900 026900 EJECT
027000 027000*****
027100 027100* VALIDATE THE INPUT PARM CODE *
027200 027200*****
027300 027300

```

- 2 Repeat the Branch Backup function to return to successive Branch locations.

Viewing Copybooks In-context

In many cases, data used by a program is defined in COBOL copybooks. In the editor, you are working with the COBOL source code. Because of this, the copybooks are not expanded. SmartEdit provides the Zoom In and Zoom Out functions to expand copybooks in the display at the point where they are used.

Adding a Copybook to the Display

To expand a copybook inline and in-context using CUA

- 1 Locate the copy statement.
- 2 Select View ► Zoom In and press Enter.
- 3 Place the cursor anywhere on the line with COPY VIAFMAST as shown in [Figure 25](#) and press Enter.

Figure 25 • Viewing Copybooks in Context

```

File View Search Check List Options Help
-----
ASG-SmartEdit -      EDIT - VIAINST.CE500001.CNTL(VIA  PLACE CURSOR ON TARGET.
Command ==> ACTION                                     Scroll ==> PAGE
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. VIAFDEMO.
000300 000300 AUTHOR. WRITTEN BY ASG AT LANGLVL 2.
000400 000400 ENVIRONMENT DIVISION.
000500 000500 CONFIGURATION SECTION.
000600 000600 SOURCE-COMPUTER. IBM-370.
000700 000700 OBJECT-COMPUTER. IBM-370.
000800 000800 INPUT-OUTPUT SECTION.
000900 000900 FILE-CONTROL.
001000 001000     SELECT MASTERIN ASSIGN TO S-MASTERIN.
001100 001100     SELECT MASTER-RPT ASSIGN TO S-MREPORT.
001200 001200 DATA DIVISION.
001300 001300 FILE SECTION.
001400 001400 FD  MASTERIN
001500 001500     RECORDING MODE IS F
001600 001600     BLOCK CONTAINS 0 RECORDS
001700 001700     LABEL RECORDS ARE STANDARD.
001800 001800
001900 001900     COPY VIAFMAST.
002000 002000

```

[Figure 26](#) shows how SmartEdit retrieves the source for the selected copybook from your copy library and expands it in the display of the source.

Figure 26 • Sample Copybook Expansion

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==> Scroll ==> PAGE
001200 001200 DATA DIVISION.
001300 001300 FILE SECTION.
001400 001400 FD MASTERIN
001500 001500 RECORDING MODE IS F
001600 001600 BLOCK CONTAINS 0 RECORDS
001700 001700 LABEL RECORDS ARE STANDARD.
001800 001800
001900 001900 COPY VIAFMAS.
***** LINES COPIED FROM VIAINST.CE50D001.CNTL *****
000100 01 MASTER-IN.
000200 05 CLIENT-ID.
000300 10 DISTRICT-ID PIC 9(3).
000400 10 CUSTOMER-ID PIC 9(3).
000500 05 NAME PIC X(24).
000600 05 ADDRESS1 PIC X(24).
000700 05 CITY PIC X(20).
000800 05 STATE PIC X(2).
000900 05 ZIP.
001000 10 ZIP-CODE PIC 9(5).
001100 10 FILLER PIC 9(11).

```

The copybook source lines are displayed with =NOTE= in the line command area indicating that these lines are for viewing only and cannot be updated within this edit session.

Use the standard ISPF editor commands or PF keys to scroll the display to see all lines in the copybook.

If the library containing the copybook member is not available, SmartEdit displays an error message. Select Options ► Refresh to access the Options - COPY/Include Libraries screen (see ["Defining Options - COPY/Include Libraries" on page 24](#) for more information). Add the appropriate COPY libraries to the list of libraries on this screen.

You can also use the Zoom In function to retrieve and expand these statements using the appropriate file, dictionary, or catalog:

```

PANVALET ++INCLUDE
LIBRARIAN -INC
DATACOM COPYDD
IDMS COPY IDMS
SQL EXEC SQL INCLUDE

```

Removing the Copybook from the Display

After you have reviewed the contents of a copybook and no longer need to refer to it, you can remove it from the display using the Zoom Out function.

To remove the copybook from the display

- 1 Select View ► Zoom Out and press Enter to display the screen shown in [Figure 27](#).

Figure 27 • Zoom Out Example

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE500001.CNTL(VIA PLACE CURSOR ON TARGET.
Command ==> ACTION Scroll ==> PAGE
001200 001200 DATA DIVISION.
001300 001300 FILE SECTION.
001400 001400 FD MASTERIN
001500 001500 RECORDING MODE IS F
001600 001600 BLOCK CONTAINS 0 RECORDS
001700 001700 LABEL RECORDS ARE STANDARD.
001800 001800
001900 001900 COPY VIAFMAST.
          ***** LINES COPIED FROM VIAINST.CE500001.CNTL *****
000100 01 MASTER-IN.
000200 05 CLIENT-ID.
000300 10 DISTRICT-ID PIC 9(3).
000400 10 CUSTOMER-ID PIC 9(3).
000500 05 NAME PIC X(24).
000600 05 ADDRESS1 PIC X(24).
000700 05 CITY PIC X(20).
000800 05 STATE PIC X(2).
000900 05 ZIP.
001000 10 ZIP-CODE PIC 9(5).
001100 10 FILLER PIC 9(11).

```

- 2 Place the cursor on the line with COPY VIAFMAST, or any line of the copybook, and press Enter. [Figure 28](#) shows how the expanded copybook is removed from the display.

Figure 28 • Results of Zoom Out Function

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE500001.CNTL(VIA COPY LINE(S) REMOVED
Command ==> Scroll ==> PAGE
001200 001200 DATA DIVISION.
001300 001300 FILE SECTION.
001400 001400 FD MASTERIN
001500 001500 RECORDING MODE IS F
001600 001600 BLOCK CONTAINS 0 RECORDS
001700 001700 LABEL RECORDS ARE STANDARD.
001800 001800
001900 001900 COPY VIAFMAST.
002000 002000
002100 002100 FD MASTER-RPT
002200 002200 RECORDING MODE IS F
002300 002300 BLOCK CONTAINS 0 RECORDS
002400 002400 LABEL RECORDS ARE STANDARD.
002500 002500 01 MAST-RPT.
002600 002600 05 FILLER PIC X(133).
002700 002700
002800 002800 WORKING-STORAGE SECTION.
002900 002900 77 FIRST-TIME PIC X VALUE 'Y'.
003000 003000 77 MASTER-END-OF-FILE PIC X VALUE SPACES.
003100 003100 88 END-INPUT VALUE 'X'.

```

The Zoom In and Zoom Out options have corresponding line commands. Use these by typing ZI on the line containing the COBOL COPY statement you want to expand and ZO beside any line of the copybook you want to remove from the display.

The View pull-down provides an option to expand all copybooks at one time. To use this function, select View ► Zoom All Copies. You can use Zoom Out to remove individual copybooks from the display or Reset to remove all copybooks at one time.

Checking Program Syntax

The SmartEdit Check function provides an online syntax check mechanism for the COBOL source currently being edited. The Check function performs a syntax check of the program, and adds error messages to the display directly below the line where the errors are encountered.

Compiler information and options are specified at the time that SmartEdit is installed. You can change the compiler dataset name and the literal delimiter (APOST or QUOTE) using the ["Check - Modules and Libraries Pop-up" on page 144](#) and the ["Check - Compiler Options Pop-up" on page 140](#), respectively. Change Copy libraries on the Options - COPY/Include Libraries screen. Change compiler options, preprocessors, and libraries on the Check - Parameters pop-up.

Note: _____

If the program to be checked contains CICS, DLI, DB2, CICS and DLI, or CICS and DB2, then specify CICS, DLI, and/or DB2 on the Check - Parameters pop-up.

Checking for Syntax Errors

To check for syntax errors using CUA, follow this step:

- Select Check ► Perform Syntax Check and press Enter. The screen shown in [Figure 29](#) displays.

Figure 29 • Syntax Checking Example

```

File View Search Check List Options Help
-----
ASG-SmartEdit -      EDIT - VIAINST.CE50D001.CNTL(VIA      1 ERROR(S) FOUND
Command ==>          Scroll ==> PAGE
023100 023100      IF DEBUG-PARM = 'TEST'
023200              READY TRACE.
-          IGYPS2021-U 'READY TRACE', 'RESET TRACE' AND 'SERVICE RELOAD' ARE NO
              LONGER SUPPORTED. A 'CONTINUE' STATEMENT WAS ASSUMED.  1 3 3
023300 023300
023400 023400      PERFORM P005-VAL-PARM
023500 023500              THRU P005-EXIT.
023600 023600
023700 023700      PERFORM P010-OPEN
023800 023800              THRU P019-EXIT.
023900 023900
024000 024000      PERFORM P155-CL-SUBTOT
024100 024100              THRU P159-EXIT.
024200 024200
024300 024300      PERFORM P120-READ
024400 024400              THRU P129-EXIT.
024500 024500
024600 024600      MOVE ZIP-CODE TO HLD-ZIP.
024700 024700      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX.
024800 024800      MOVE 1 TO PAGE-CNT.

```

Error messages resulting from the syntax check are displayed immediately following the statement containing the error. The display is positioned at the first error with the number of errors found displayed in the short message area.

Locating Errors from the Check Function

After a syntax check, the display is positioned at the first error found.

To locate additional errors in a program

- 1 Select Search ► Subset and press Enter. The Search - COBOL Subset Name pop-up, shown in [Figure 30](#), displays.

Figure 30 • Search - COBOL Subset Name Pop-up

Search - COBOL Subset Name

Type a subset name and select search options. Then press Enter. For a selection list, type a pattern (e.g. ABC*) in the name area.

Subset Name _____

Direction	Options	Action
1. All	- IN-Clause...	1. Find
2. Next		2. Highlight
3. Previous		3. Scroll
4. First		4. Print
5. Last		5. Punch
		6. Exclude

- 2 Type HIGHLIGHT in the Subset Name field, select Next in the Direction field to move down in the source code, select Scroll in the Action field to specify that the display should be positioned to the error text found, and press Enter.
- 3 At the prompt, position the cursor below the current error messages on the display and press Enter.

Note:

Repeat [step 1](#) through [step 3](#) to scroll through all the errors found by the syntax check function. As an alternative to these steps, you can use the SCROLL command to locate the syntax errors. Type SCROLL HIGH NEXT and position the cursor after the errors shown on the display.

After using [step 1](#) through [step 3](#), or the SCROLL command, you can use RSCROLL to find succeeding errors. For convenience, RSCROLL can be assigned to a PF key.

The CHECK command uses temporary files for its processing. If CHECK fails due to insufficient temporary file space, then increase the size of the Work File on the Options - Product Allocations pop-up. You can display the Options - Product Allocations pop-up by selecting Options ► Product Allocations. See ["Options - Product Allocations Pop-up" on page 164](#)) and the ["CHECK Command" on page 194](#) for more information.

Using the COBOL Intelligent Search Feature

This section provides descriptions and examples of how to use the COBOL intelligent Search feature to find these items:

- Data relationships and references
- Data item size change impact
- Data usage ripple effect impact
- Related COBOL verb subsets

COBOL Intelligent Search Description

The Search function provides an automated COBOL-sensitive search that performs these functions:

- Displays all references by tagging and highlighting the uses (USE), modifications (MOD), and definitions (DEF) of all related data fields.
- Finds all related data (i.e., aliases, renames, redefines) for a specified data field.
- Locates COBOL verbs and subsets (i.e., COBOLII, CICS, I/O, DL/I, SQL, IDMS, COBOL370, and Exits).
- Provides an overview of the program structure (i.e., divisions, structures, and perform ranges).

The results of a Search are used for these functions:

- Scroll through the highlighted lines using the SCROLL and RSCROLL commands.
- Group the highlighted lines together without intervening source code using the EXCLUDE command, X NHI.
- Search for another target in the highlighted lines.
- Save the results for later reference using the Print and Punch actions within the Search pop-ups. You can also use the LPRINT and LPUNCH commands.

You can search the entire program or use the IN options to search only for targets in statements belonging to a COBOL subset. The result of the search consists of all the lines that contain the target. The lines are highlighted and the cursor is placed beneath the first occurrence of the target. Targets that were highlighted as the result of a previous command are reset so that only the results of the current command are highlighted.

SmartEdit can search for complex items that would take many separate searches if done manually.

Finding Data Relationships and References

To locate all references to the data item LOAN-AMT

- 1 Select Search ► Data (to indicate that the target must be a data item) and press Enter. The Search - Data Name pop-up, shown in [Figure 31](#), displays.

Figure 31 • Search - Data Name Pop-up - Intelligent Search

Search - Data Name

Type a data name and select search options. Then press Enter. For a selection list, enter a pattern (e.g. ABC*) in the name area.

Data Name LOAN-AMT_

Data References 1 1. All 2. Defs 3. Uses 4. Mods	Indirect Impact 1 1. None 2. Of Size Change 3. Of Value Change	Size change levels . . . ---
---	--	--

Direction 1 1. All 2. Next 3. Previous 4. First 5. Last	Options - No Data Aliasing - IN-Clause...	Action 1 1. Find 2. Highlight 3. Scroll 4. Print 5. Punch 6. Exclude
---	--	---

- 2 Type LOAN-AMT in the Data Name field and press Enter.

As shown in [Figure 32](#), the resulting display shows all definitions (DEF), uses (USE), and modifications (MOD) to the data item LOAN-AMT.

Figure 32 • Intelligent Data Item Search Example

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==>
ASG00640I 62 DATA DEFS, 31 LEVELS, FOUND FOR SUB-LOAN-AMT.
- - - - - 15 Line(s) not Displayed
001400 FD MASTERIN
001500 001500 RECORDING MODE IS F
001600 001600 BLOCK CONTAINS 0 RECORDS
001700 001700 LABEL RECORDS ARE STANDARD.
001800 001800
001900 COPY VIAFMAS.
002000 002000
002100 FD MASTER-RPT
002200 002200 RECORDING MODE IS F
002300 002300 BLOCK CONTAINS 0 RECORDS
002400 002400 LABEL RECORDS ARE STANDARD.
002500 01 MAST-RPT.
002600 05 FILLER
- - - - - PIC X(133).
- - - - - 10 Line(s) not Displayed
003700 01 AUG-AMT
- - - - - PIC 9(8)V99 COMP.
- - - - - 5 Line(s) not Displayed
004300 01 DETAIL-LINE1.
- - - - - 4 Line(s) not Displayed

```

The COBOL intelligence designed into SmartEdit allows it to find references to the specified data item inside copybooks and group levels. This provides a much more useful tool for the COBOL programmer than standard editor text searching.

MASTER-IN is the 01 level in VIAFMAST for the data field LOAN-AMT. The display is positioned at the first reference to LOAN-AMT in the PROCEDURE DIVISION.

The DEF==> at the source line COPY VIAFMAST indicates that there are one or more definitions of LOAN-AMT inside this copybook. Use the Zoom In function to find the detail of the definitions inside the copybook.

The REF==> at the CALL statement indicates a MOD and USE combined.

Note:

This example assumes that the option XMODE is ON so that only lines that contain the target are displayed. XMODE is set using the Options - Processing Modes pop-up, which you can access by selecting Options ► Processing Modes (see ["Options - Processing Modes Pop-up" on page 169](#) for more information).

Finding the Impact of Changing a Data Field Size

The program maintenance activity frequently requires changing the size of a data field. The SmartEdit intelligent Search function helps determine the impact of changing the size of a data field by locating all data items in the program that are affected directly or indirectly.

To locate the definitions of all data items that may be impacted by a change in the size of the data field SUB-LOAN-AMT

- 1 Select Search ► Data and press Enter. The Search - Data name pop-up, shown in [Figure 33 on page 49](#), displays.

- 2 Type SUB-LOAN-AMT as the Data Name, select Defs in the Data References field, Of Size Change in the Indirect Impact field (so that all data items affected by size changes are located), and press Enter.

Figure 33 • Searching for the Impact of Size Change

```

Search - Data Name

Type a data name and select search options. Then press Enter. For a
selection list, enter a pattern (e.g. ABC*) in the name area.

Data Name SUB-LOAN-AMT

Data References      Indirect Impact      Size change
2 1. All             2 1. None          levels . . . ---
  2. Defs            2 2. Of Size Change
  3. Uses             3. Of Value Change
  4. Mods

Direction           Options              Action
1 1. All             - No Data Aliasing      1 1. Find
  2. Next             - IN-Clause...          2. Highlight
  3. Previous          3. Scroll
  4. First             4. Print
  5. Last              5. Punch
                      6. Exclude

```

The resulting display, shown in [Figure 34](#), lists all data items affected by a change in the size of SUB-LOAN-AMT. It shows 62 definitions spanning 31 levels of indirect reference.

Figure 34 • Data Items Affected by Change in Size

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==>
ASG00640I 62 DATA DEFS, 31 LEVELS, FOUND FOR SUB-LOAN-AMT.
- - - - - 15 Line(s) not Displayed
001400 FD MASTERIN
001500 001500 RECORDING MODE IS F
001600 001600 BLOCK CONTAINS 0 RECORDS
001700 001700 LABEL RECORDS ARE STANDARD.
001800 001800
001900 COPY VIAFMAST.
002000 002000
002100 FD MASTER-RPT
002200 002200 RECORDING MODE IS F
002300 002300 BLOCK CONTAINS 0 RECORDS
002400 002400 LABEL RECORDS ARE STANDARD.
002500 01 MAST-RPT.
002600 05 FILLER PIC X(133).
- - - - - 10 Line(s) not Displayed
003700 01 AUG-AMT PIC 9(8)U99 COMP.
- - - - - 5 Line(s) not Displayed
004300 01 DETAIL-LINE1.
- - - - - 4 Line(s) not Displayed

```

If you had selected All in the Data References field, the Definitions, Uses, and Modifications would be shown on the resulting display.

The SmartEdit intelligent Search function makes it easy for you to identify all of the data items in the program affected by a change in data field size.

Determining the Impact of Changing the Usage of a Data Item

When modifying the logic of a program during maintenance, you may need to change the logical use of a data item. When this change is made, you may impact related program logic, creating a ripple effect. For example:

Data item A currently allows two values. You change the code to permit three values. If the value in A is transferred to other data items and an unexpected value is found, the program may fail. The SmartEdit intelligent Search function locates those indirectly related data items and the program logic that may depend on the values contained in data item A.

The intelligent Search function with the Indirect Impact selection Of Value Change actually follows the data flow from one field to the next (Field A to B, B to C, C to D, and so forth).

To locate all data items impacted by a change in the logical usage of the data item SUB-LOAN-CNT

- 1 Select Search ► Data and press Enter. The Search - Data Name pop-up, shown in [Figure 35](#), displays.

Figure 35 • Searching for Impact of Data Item Value Change

Search - Data Name

Type a data name and select search options. Then press Enter. For a selection list, enter a pattern (e.g. ABC*) in the name area.

Data Name SUB-LOAN-CNT

Data References 1 1. All 2 2. Defs 3 3. Uses 4 4. Mods	Indirect Impact 3 1. None 2 2. Of Size Change 3 3. Of Value Change	Size change levels . . . ___
---	--	--

Direction 1 1. All 2 2. Next 3 3. Previous 4 4. First 5 5. Last	Options - No Data Aliasing - IN-Clause...	Action 1 1. Find 2 2. Highlight 3 3. Scroll 4 4. Print 5 5. Punch 6 6. Exclude
---	--	---

- 2 Type SUB-LOAN-CNT in the Data Name field, select All in the Data References field, Of Value Change in the Indirect Impact field, and press Enter.

The results of the dataname search, shown in [Figure 36](#), lists all data definitions (DEF), uses (USE), and modifications (MOD) impacted when the value in the SUB-LOAN-CNT data field is changed.

Figure 36 • All Items Affected by Change in Value

```

File View Search Check List Options Help
-----
ASG-SmartEdit -      EDIT - VIINST.CE50D001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==>
ASG0639I 22 DATA REFS, 9 DEFS, 1 USE, 12 MODS, FOUND FOR SUB-LOAN-CNT.
032800 032800 WRITE MAST-RPT FROM DETAIL-LINE1.
032900 032900
033000 033000 MOVE NAME TO DET-NAME.
033100 033100 MOVE YEAR-TO-DATE-INTEREST TO DET-YTD-INT.
033200 033200 MOVE LAST-BILL-DATE TO DET-LAST-BILL-DATE.
033300 033300 WRITE MAST-RPT FROM DETAIL-LINE2.
033400 033400
033500 033500 MOVE ADDRESS1 TO DET-ADDRESS.
033600 033600 MOVE AREA-CODE TO DET-AREA-CODE.
033700 033700 MOVE EXCHANGE TO DET-EXCHANGE.
033800 033800 MOVE PHONE-NUMBER TO DET-PHONE-NUMBER.
033900 033900 MOVE PAYMENT-AMT TO DET-PAYMENT-AMT.
034000 034000 MOVE LOAN-TYPE TO DET-LOAN-TYPE.
034100 034100 WRITE MAST-RPT FROM DETAIL-LINE3.
034200 034200
034300 034300 MOVE STATE TO DET-STATE.

```

Finding Subsets of Related COBOL Verbs

During a program editing session, you may need to locate all statements of a particular type, such as CALL, I/O, IDMS, SQL, or COBOL370. The SmartEdit Search function provides the capability to search for COBOL, DL/I, SQL, IMS, or IDMS language subsets as well as tags and highlighted lines.

To locate all CALL statements in a program

- 1 Select Search ► Subset and press Enter. The Search - COBOL Subset Name pop-up, shown in [Figure 37](#), displays.

Figure 37 • Search - COBOL Subset Name Pop-up

```

Search - COBOL Subset Name

Type a subset name and select search options. Then press Enter. For
a selection list, type a pattern (e.g. ABC*) in the name area.

Subset Name _____

Direction      Options      Action
1 1. All        - IN-Clause... 1 1. Find
2. Next
3. Previous
4. First
5. Last
                                2. Highlight
                                3. Scroll
                                4. Print
                                5. Punch
                                6. Exclude

```

- 2 To view a list of the subset choices and to make a selection, leave the Subset Name field blank and press Enter. The Selection List - Subset Names pop-up, shown in [Figure 38](#), displays.

Figure 38 • Selection List - Subset Names Pop-up

```

Command ==> Selection List - Subset Names LINE 1 OF 93
Scroll ==> 3

$ Subset Description
-----
- AS(signment) - COBOL: ACCEPT, ADD, SUBTRACT, MULTIPLY, DIVIDE, BY,
                  COMPUTE, MOVE, FROM, INSPECT, EXAMINE, STRING,
                  UNSTRING, SET, SEARCH, TRANSFORM.
                  IDMS: ACCEPT, CHECK, ERASE, GET, INQUIRE, LOAD, PUT,
                  RETURN.
                  SQL: FETCH, SELECT INTO, SET.
s CA(11) - COBOL: CALL, CANCEL, ENTRY, intrinsic function calls.
          CICS: LINK, XCTL.
          IDMS: ATTACH, LINK, XCTL.
- CI(cs) - EXEC CICS and EXEC DL/I commands.
- COBOLII - CALL (COBOLII), PERFORM (COBOLII), CONTINUE, SET
          (to TRUE), SERVICE LABEL, EVALUATE, INITIALIZE, END
          VERBS.
- COBOL370 - Statements and clauses unique to COBOL/370 such as
            intrinsic function calls, procedure pointers and
            calls to the LE/370 run-time environment.
- COM(ment) - Comment lines and Identification division.

```

- 3 Select CALL from the subset list by typing S beside CA(11) and pressing Enter.

The subset name is returned in the Subset Name field on the Search - COBOL Subset Name pop-up as shown in [Figure 39](#).

Figure 39 • Searching for COBOL Subsets

```

Search - COBOL Subset Name

Type a subset name and select search options. Then press Enter. For
a selection list, type a pattern (e.g. ABC*) in the name area.

Subset Name CALL

Direction Options Action
1 1. All - IN-Clause... 1 1. Find
2. Next 2. Highlight
3. Previous 3. Scroll
4. First 4. Print
5. Last 5. Punch
6. Exclude

```

- 4 Select All in the Direction field, Find in the Action field, and press Enter to execute the search request.

The resulting screen, shown in [Figure 40](#) displays all COBOL CALL statements in the program.

Figure 40 • Results of COBOL Subset Search

```

File View Search Check List Options Help
ASG-SmartEdit - EDIT - VIINST.CE50D001.CNTL(VIA) 10 LINES FOUND
Command ===> scroll ==> PAGE
025600 025600
025700 025700 CALL 'VIAFDEM1' USING MASTER-IN
025800 025800 MASTER-END-OF-FILE
025900 025900 MASTER-REPORT-DATE.
026000 026000
026100 026100 PERFORM P100-PRINT
026200 026200 THRU P119-EXIT.
026300 026300
026400 026400 PERFORM P120-READ
026500 026500 THRU P129-EXIT.
026600 026600
026700 026700 P000-EXIT.
026800 026800 EXIT.
026900 026900 EJECT
027000 027000 *****
027100 027100* VALIDATE THE INPUT PARM CODE *
027200 027200*****
027300 027300
027400 027400 P005-VAL-PARM.
027500 027500 IF DBA-DEPT-CODE > 24

```

SmartEdit classifies COBOL statements into subsets by grouping together COBOL verbs of a similar nature. For a list of COBOL language subsets, access the List action from the CUA action bar or see ["Commands" on page 181](#).

COBOL Program Understanding Features

Understanding program structure and flow is critical when modifying the program logic or changing the logical usage of a data item. SmartEdit provides functions to save you time in understanding the structure and flow of the program. These features include these View options:

- Zoom In and Zoom Out to view the program in physical order
- Paragraph X-Ref to view the program flow at the paragraph level
- Tree view to view the program in logical execution order

Understanding Program Structure

The SmartEdit Zoom In and Zoom Out functions add or remove source lines from the display according to the hierarchical levels of the program. These functions provide a view of the program in physical order and a means of stepping through each hierarchical level of the program.

SmartEdit defines these three levels of hierarchy for a program:

- The program level hierarchy:
 - The IDENTIFICATION DIVISION statement
 - The PROGRAM-ID statement
- The PROCEDURE DIVISION hierarchy:
 - Section labels
 - Paragraph labels
 - Paragraph code
- The DATA DIVISION hierarchy:
 - Sections
 - FDs, 01, or 77 levels
 - All definitions within an 01 level

The process of using the Zoom In and Zoom Out functions begins by excluding all program source lines from the display and then performing successive Zoom In functions to reveal the program structure in stages.

To exclude all lines from the display, follow this step:

- Type EXCLUDE ALL. [Figure 41](#) shows that all lines in the program have been excluded from view.

Figure 41 • Result with all Lines Excluded

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIINST.CE50D001.CNTL(VIA All lines excluded
Command ==> Scroll ==> PAGE
***** Top of Data *****
- - - - - 474 Line(s) not Displayed
***** Bottom of Data *****

```

Note:

You can abbreviate the EXCLUDE ALL command using X. The operand ALL is the default. The Exclude function is also available by selecting View ► Exclude.

Revealing Program Structure Using the ZOOMIN Function

Starting with all lines excluded from the display, begin to reveal the program structure by displaying the program level in the structure hierarchy.

To use the ZOOMIN command to reveal program structure, follow this step:

- Type ZOOMIN on the command line and press Enter. [Figure 42](#) shows the program level hierarchy.

Figure 42 • Program Level Hierarchy

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - UIAINST.CE50D001.CNTL(VIA 2 LINES UNEXCLUDED
Command ==> Scroll ==> PAGE
***** Top of Data *****
- - - - - 2 Line(s) not Displayed
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. UIAFDEMO.
- - - - - 470 Line(s) not Displayed
***** Bottom of Data *****

```

Note:

You can assign the ZOOMIN (ZI) command to a PF key for ease of use in revealing program structure.

To add the next level of the hierarchy to the display

- 1 Type ZOOMIN on the command line and press Enter. The COBOL division headers have been added to the display, as shown in [Figure 43](#).

Figure 43 • Second Level of Program Structure

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - UIAINST.CE50D001.CNTL(VIA 3 LINES UNEXCLUDED
Command ==> Scroll ==> PAGE
***** Top of Data *****
- - - - - 2 Line(s) not Displayed
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. UIAFDEMO.
- - - - - 1 Line(s) not Displayed
000400 000400 ENVIRONMENT DIVISION.
- - - - - 7 Line(s) not Displayed
001200 001200 DATA DIVISION.
- - - - - 194 Line(s) not Displayed
020700 020700 PROCEDURE DIVISION.
- - - - - 265 Line(s) not Displayed
***** Bottom of Data *****

```

The message in the short message area indicates that three additional lines of the program were unexcluded (added to the display) as the result of the Zoom In action.

- 2 You may continue to add hierarchical levels of the PROCEDURE DIVISION structure, as shown in [Figure 44](#), by executing the ZOOMIN command and positioning the cursor on the PROCEDURE DIVISION header. Repeat this step to add SECTIONS and then paragraphs to the display until the entire physical structure is revealed. Remember to position the cursor on the PROCEDURE DIVISION header each time.

Figure 44 • Paragraphs within the PROCEDURE DIVISION

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIA 25 LINES UNEXCLUDED
Command ==> Scroll ==> PAGE
***** Top of Data *****
- - - - - 2 Line(s) not Displayed
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. VIAFDEMO.
- - - - - 1 Line(s) not Displayed
000400 000400 ENVIRONMENT DIVISION.
- - - - - 7 Line(s) not Displayed
001200 001200 DATA DIVISION.
- - - - - 194 Line(s) not Displayed
020700 020700 PROCEDURE DIVISION.
- - - - - 21 Line(s) not Displayed
022900 022900 PROGRAM-INIT.
- - - - - 25 Line(s) not Displayed
025500 025500 P000-NEXT.
- - - - - 11 Line(s) not Displayed
026700 026700 P000-EXIT.
- - - - - 6 Line(s) not Displayed
027400 027400 P005-VAL-PARM.
- - - - - 5 Line(s) not Displayed
028000 028000 P005-EXIT.

```

- 3 Type ZOOMIN , position the cursor on the PROCEDURE DIVISION statement once more, and press Enter to add the source statements to the display. [Figure 45](#) shows the result of this command.

Figure 45 • Source Statements within PROCEDURE DIVISION

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIA 239 LINES UNEXCLUDED
Command ==> Scroll ==> PAGE
***** Top of Data *****
- - - - - 2 Line(s) not Displayed
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. VIAFDEMO.
- - - - - 1 Line(s) not Displayed
000400 000400 ENVIRONMENT DIVISION.
- - - - - 7 Line(s) not Displayed
001200 001200 DATA DIVISION.
- - - - - 194 Line(s) not Displayed
020700 020700 PROCEDURE DIVISION.
020800 020800
020900 020900*****
021000 021000* PERFORM PROGRAM INITIALIZATION *
021100 021100*****
021200 021200
021300 021300 PERFORM PROGRAM-INIT.
021400 021400
021500 021500*****
021600 021600* DRIVER SUBROUTINE LOOP *
021700 021700*****

```

Removing Parts of the Program Structure from the Display

To retrace the steps through the program structure using the ZOOMOUT primary command, follow this step:

- Type ZOOMOUT on the command line and press Enter.

As shown in [Figure 46](#), the statements for the PROCEDURE DIVISION have been removed from the display.

Figure 46 • Removing Levels of the Hierarchy from the Display

```

File View Search Check List Options Help
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL(VIA 25 LINES UNEXCLUDED
Command ==> Scroll ==> PAGE
***** Top of Data *****
000100 000100 IDENTIFICATION DIVISION.
000200 000200 PROGRAM-ID. VIAFDEMO.
000400 000400 ENVIRONMENT DIVISION.
001200 001200 DATA DIVISION.
020700 020700 PROCEDURE DIVISION.
022900 022900 PROGRAM-INIT.
025500 025500 P000-NEXT.
026700 026700 P000-EXIT.
027400 027400 P005-VAL-PARM.
028000 028000 P005-EXIT.

```

Each execution of the ZOOMOUT function removes one hierarchical level of the program from the display. Repeat the ZOOMOUT command until you reach the desired level in the hierarchy.

Understanding Program Flow of Control

When modifying program logic, understanding the flow of control in the program is very important. SmartEdit provides an interactive paragraph cross-reference function to assist in gaining this understanding.

To view all the paragraphs that transfer control to paragraph P120-READ

- 1 Select View ► Paragraph X-Ref and press Enter. The View - Paragraph Cross Reference Request pop-up, shown in [Figure 47 on page 58](#), displays.

- 2 Type P120-READ in the Target Label field on the View - Paragraph Cross Reference Request pop-up. So that the display shows all the paragraphs that transfer control to the specified target label, select Previous in the Direction field and press Enter.

Figure 47 • Selecting Paragraph Cross Reference Display

```

View - Paragraph Cross Reference Request

Type Target Name(s) and select a Direction. Then press Enter.
For a selection list, type a pattern in the Target field (except
for Line Range).

Target                                     Direction
Use cursor                               1  1. Previous
Program . . .                             2. Next
Subset . . .
Label . . . P120-READ
Line Range

```

[Figure 48](#) shows the results of the paragraph cross-reference request.

Figure 48 • Result of Paragraph Cross Reference

```

View - Paragraph Cross Reference
Command ==> _ Scroll ==> 3
Target(s): P120-READ
Direction: Previous

A : Add to target      P : Execute PREF      $ : Select for viewing

-----
Comes From              How              Target Paragraph(s)
-----
- PROGRAM-INIT          PERFORM          P120-READ
- P000-NEXT              PERFORM          "  "
- P120-READ              GOTO             "  "
- P119-EXIT              PREV PARA        "  "
***** BOTTOM OF DATA *****

```

You can access the paragraph P120-READ from these areas:

- The PERFORM statement in PROGRAM-INIT.
- Another PERFORM statement in P000-NEXT.
- A P120-READ that has a GO TO that returns to itself.
- The potential fallthru from the previous paragraph P119-EXIT. The PREV PARA means that a fallthru can occur at run-time.

For an explanation of the fields on the View - Paragraph Cross Reference pop-up, see the definitions in ["Fields" on page 61](#).

You can continue to follow program logic by adding another target paragraph to the Paragraph Cross Reference display.

To locate the paragraphs that transfer control to P000-NEXT

- 1 Type A in the selection field beside the P000-NEXT paragraph and press Enter. This paragraph is added to the screen.

[Figure 49](#) displays the cross-reference for paragraphs P000-NEXT and P120-READ. In this example, the PROCEDURE DIVISION PERFORMs P000-NEXT and P000-NEXT PERFORMs P120-READ. There is also a potential fallthru to P000-NEXT, (PREV PARA), from PROGRAM-INIT.

Figure 49 • Adding a Paragraph to the Display

View - Paragraph Cross Reference		
Command ==> _____ Scroll ==> 3		
Target(s): P000-NEXT + P120-READ		
Direction: Previous		
A : Add to target P : Execute PREF \$: Select for viewing		
Comes From	How	Target Paragraph(s)
- PROCEDURE DIVISION	PERFORM	P000-NEXT
- PROGRAM-INIT	PREV PARA	" "
- " "	PERFORM	P120-READ
- P000-NEXT	PERFORM	" "
- P120-READ	GOTO	" "
- P119-EXIT	PREV PARA	" "
***** BOTTOM OF DATA *****		

- 2 Type END or press PF3 to return to the editor screen.

Viewing Complete Paragraph Cross-references

To view a complete paragraph cross-reference

- 1 Select View ► Paragraph X-Ref and press Enter. The View - Paragraph Cross Reference pop-up, shown in [Figure 50](#), displays.

Figure 50 • View - Paragraph Cross Reference Pop-up

```

View - Paragraph Cross Reference
Command ==> _____ Scroll ==> 3

Target(s): VIAFDEMO
Direction: Next

A : Add to target      P : Execute PREF      S : Select for viewing

Target Paragraph(s)    How      Goes To
-----
PROCEDURE DIVISION    PERF,NEXT    PROGRAM-INIT
" "                  PERFORM      P000-NEXT
" "                  PROGRAM EXIT
" "                  NEXT PARA    PROGRAM-INIT
PROGRAM-INIT          NEXT PARA    P000-NEXT
" "                  PERFORM      P005-VAL-PARM
" "                  PERFORM      P010-OPEN
" "                  PERFORM      P120-READ
" "                  PERFORM      P155-CL-SUBTOT
P000-NEXT             NEXT PARA
" "                  PERFORM      P100-PRINT
" "                  PERFORM      P120-READ

```

- 2 Type VIAFDEMO in the Target Program field, Next in the Direction field (so that the display shows all the paragraphs to which they transfer control), and press Enter.

Typing Next in the Direction field results in the display of all PROCEDURE DIVISION paragraphs and section names along with the paragraph or section name(s) to which they transfer control.

- 3 Type END or press PF3 to return to the editor screen.

Note:

You can use the standard editor commands, UP and DOWN, to scroll the complete cross-reference results.

To view the source code for any paragraph on the Paragraph X-Ref display, type S to the left of the paragraph name and press Enter. This returns you to the editor with the display positioned at that paragraph.

For COBOL II Release 3 and later programs that contain subprograms, the Paragraph Cross Reference screen also includes CALLS and CALL RETN as forms of transferring control to or from a paragraph.

Fields

Field	Description
TARGET(S)	Specifies the paragraph name(s) of the requested target.
DIRECTION	Specifies the type of transfer. PREVIOUS highlights paragraphs from which control is transferred. NEXT highlights paragraphs to which control is transferred. The default is PREVIOUS.
Line command area	<p>Specifies the line command area, which is the field to the left of the COMES FROM field. These are the valid entries:</p> <ul style="list-style-type: none"> • A - Adds the selected paragraph to the target paragraph(s). If the direction is PREVIOUS, the target paragraph is added at the top of the list. If the direction is NEXT, the target paragraph is added to the end of the list. • P - Executes another paragraph cross-reference. • S - Returns to the source code with the first line of the paragraph displayed at the top of the screen.
COMES FROM	Lists the paragraph names that transfer control to the target paragraph(s). All paragraphs that relate to the target paragraph are grouped together.

Field	Description																												
HOW	<p>Indicates how control is transferred to or from the target paragraph. These are the valid values:</p> <table> <tr> <th>Value</th><th>Abbreviation</th></tr> <tr> <td>PERFORM</td><td>PERF(3),PERF,P(2),P</td></tr> <tr> <td>GOTO</td><td>GO(3), G(2), G</td></tr> <tr> <td>ALTERED GOTO</td><td>ALTER, AG</td></tr> <tr> <td>PREV PARA</td><td>PREV</td></tr> <tr> <td>NEXT PARA</td><td>NEXT</td></tr> <tr> <td>GOTO DEPENDING</td><td>GODEP, GD</td></tr> <tr> <td>FALLTHRU</td><td>FALL, F</td></tr> <tr> <td>USE ON ERROR</td><td>USE ERROR, U</td></tr> <tr> <td>USE FOR DEBUG</td><td>USE DEBUG, U</td></tr> <tr> <td>USE BEFORE REPORT</td><td>USE REPORT, U</td></tr> <tr> <td>*CALL INTERNAL</td><td>CLL INT, CI</td></tr> <tr> <td>*CALL RETURN</td><td>CLL RTN, CR</td></tr> <tr> <td colspan="2">*COBOL II Release 3 and later programs containing subprograms</td></tr> </table>	Value	Abbreviation	PERFORM	PERF(3),PERF,P(2),P	GOTO	GO(3), G(2), G	ALTERED GOTO	ALTER, AG	PREV PARA	PREV	NEXT PARA	NEXT	GOTO DEPENDING	GODEP, GD	FALLTHRU	FALL, F	USE ON ERROR	USE ERROR, U	USE FOR DEBUG	USE DEBUG, U	USE BEFORE REPORT	USE REPORT, U	*CALL INTERNAL	CLL INT, CI	*CALL RETURN	CLL RTN, CR	*COBOL II Release 3 and later programs containing subprograms	
Value	Abbreviation																												
PERFORM	PERF(3),PERF,P(2),P																												
GOTO	GO(3), G(2), G																												
ALTERED GOTO	ALTER, AG																												
PREV PARA	PREV																												
NEXT PARA	NEXT																												
GOTO DEPENDING	GODEP, GD																												
FALLTHRU	FALL, F																												
USE ON ERROR	USE ERROR, U																												
USE FOR DEBUG	USE DEBUG, U																												
USE BEFORE REPORT	USE REPORT, U																												
*CALL INTERNAL	CLL INT, CI																												
*CALL RETURN	CLL RTN, CR																												
*COBOL II Release 3 and later programs containing subprograms																													
TARGET PARAGRAPH(S)	<p>Specifies the paragraphs requested as the target in the PREF command. Ditto marks (") indicate multiple paragraphs transferring control to the same paragraph.</p>																												

Gaining High-level Program Understanding

Understanding the logical execution order of the program is critical to making logic changes correctly.

To display program execution order

- 1 Select View ► Tree to view the first level of the program view. The View - Tree View Request pop-up, shown in [Figure 51](#), displays.

Figure 51 • View - Tree View Request Pop-up

```

View - Tree View Request

Type request options. Then press Enter. For a (name) selection
list, type a pattern (or blanks) in the name field.

Levels 1
Contents
1 1. Labels
   2. Labels and performs
   3. Labels, performs and GOTOs
   4. Labels, performs, GOTOs and
     conditionals
Range
1 1. Procedure division
   2. All entry points
   3. Perfrange name
   4. Nested program name

Name -----

Options
/ Repetition

```

- 2 Keep the default selections for Levels 1 and Contents of Labels and press Enter.

The Tree View screen, shown in [Figure 52](#), displays the highest level of the program structure (the program Procedure Division). The message indicates the number of levels currently displayed and the total number of levels in the program.

Figure 52 • Result of First Level Program Display

```

File View Search Options Help
-----
ASG                                Tree View                                LINE 1 OF 2
Command ==>                        Scroll ==> 3
ASG00091 1 OF 5 LEVELS DISPLAYED IN TREEVIEW.
***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001 + PROCEDURE DIVISION.
***** ***** BOTTOM OF DATA *****

```

In the Tree View display, Levels are defined by the nesting of PERFORM statements. Indentation on the display denotes new levels.

A plus sign (+) in front of a statement indicates that there are additional structure levels contained in this label that have not been displayed.

- 3 Type ZI on the line containing PROCEDURE DIVISION to view additional levels of the program structure. You can also position the cursor on this line and select View ► Zoom In.

The Tree View screen, shown in [Figure 53](#), displays the next level in the program hierarchy, specifically, the highest level paragraphs executed in the PROCEDURE DIVISION.

Figure 53 • Next Level of Program View

```

File View Search Options Help
-----
ASG                                     Tree View                               LINE 1 OF 5
Command ==> _____ $Scroll ==> 3

***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001      PROCEDURE DIVISION.
000002 +      PROGRAM-INIT.
000002 +      P000-NEXT.
000002 +      P000-EXIT.
***** ***** BOTTOM OF DATA *****

```

The PROGRAM-INIT paragraph has a plus sign (+) indicating that lower level paragraphs exist. Use Zoom In to view the next hierarchical level of the program.

- 4 Type ZI on the line containing PROGRAM-INIT and press Enter. The Tree View screen, shown in [Figure 54](#), displays.

Figure 54 • Next Level within PROGRAM-INIT

```

File View Search Options Help
-----
ASG                                     Tree View                               LINE 1 OF 12
Command ==> _____ $Scroll ==> 3

***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001      PROCEDURE DIVISION.
000002      PROGRAM-INIT.
000003 +      P005-VAL-PARM.
000003      P005-EXIT.
000003      P010-OPEN.
000003      P019-EXIT.
000003 +      P155-CL-SUBTOT.
000003 +      P120-READ.
000003      P129-EXIT.
000002 +      P000-NEXT.
000002      P000-EXIT.
***** ***** BOTTOM OF DATA *****

```

- 5 Type ZI on the line containing P005-VAL-PARM and press Enter. [Figure 55](#) shows that paragraph P999-ABEND-PROGRAM is invoked twice by P005-VAL-PARM.

Figure 55 • Next Level within P005-VAL-PARM

```

File View Search Options Help
-----
ASG .                                     Tree View                               LINE 1 OF 14
Command ==> _____ Scroll ==> 3

***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001   PROCEDURE DIVISION.
000002   PROGRAM-INIT.
000003   P005-VAL-PARM.
000004   P999-ABEND-PROGRAM.
000004   P999-ABEND-PROGRAM.
000003   P005-EXIT.
000003   P010-OPEN.
000003   P019-EXIT.
000003 +   P155-CL-SUBTOT.
000003 +   P120-READ.
000003   P129-EXIT.
000002 +   P000-NEXT.
000002   P000-EXIT.
***** ***** BOTTOM OF DATA *****

```

Viewing Complete Program

To quickly display all of the program structure levels

- 1 Select View ► Levels and press Enter.
- 2 Change the Levels 1 to Levels MAX and press Enter. The results of a complete program view are shown in [Figure 56](#).

Figure 56 • Displaying all Levels of the Program

```

File View Search Options Help
-----
ASG                                     Tree View                               LINE 1 OF 44
Command ==> _____ Scroll ==> 3
ASG0909I 5 OF 5 LEVELS DISPLAYED IN TREEVIEW.
***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001   PROCEDURE DIVISION.
000002   PROGRAM-INIT.
000003   P005-VAL-PARM.
000004   P999-ABEND-PROGRAM.
000004   P999-ABEND-PROGRAM.
000003   P005-EXIT.
000003   P010-OPEN.
000003   P019-EXIT.
000003   P155-CL-SUBTOT.
000004   P159-EXIT.
000003   P120-READ.
000004   P170-FINAL.
000005   P999-ABEND-PROGRAM.
000004   P179-EXIT.
000004   P200-CLOSE.
000005   P129-EXIT.
000004   P129-EXIT.

```

The View pull-down on the Tree View screen contains selections that are valid while in Tree view. Type MAX in the Levels field on the View - Tree View Request pop-up shown in [Figure 75 on page 91](#) to display all program structure levels on the initial request.

Program View Tags

The Tree function generates tags to the right of lines of special interest to allow greater understanding of program flow. These are the most common tags:

- The COND tag indicates that the control flow verb is part of a conditional statement, for example IF, AT END, IF...ELSE.
- The RECUR tag indicates that the PERFORM is recursive and performs itself.
- The SOURCE (SRC) tag is placed on actual executable source lines when displayed.
- The REPEATED/REP tag indicates that a PERFORM range is repeated and the Repetition option on the View - Tree View Request pop-up was not selected.

The short form of any tag (i.e., SRC/MOD) is used when there is more than one tag to be displayed on any line. For details on the Program View tags, see ["View" on page 87](#).

Viewing Source Statements

With the execution order of the program displayed, you can view the source statements for a particular paragraph using the SmartEdit Zoom In function.

To view source statements, follow this step:

- ▶ Type ZI in the line command area for PROGRAM-INIT to display the source code for that paragraph. The source statements for the selected paragraph are added to the display, as shown in [Figure 57](#).

Figure 57 • Viewing Source Statements in a Paragraph

```

File View Search Options Help
-----
ASG                               Tree View                               LINE 1 OF 69
Command ==>                      Scroll ==> 3

***** ***** TOP OF DATA *****
000000 UIAFDEMO.
000001   PROCEDURE DIVISION.
000002     PROGRAM-INIT.
000002                                     SOURCE
000002                                     SOURCE
000002       IF DEBUG-PARM = 'TEST'
000002         READY TRACE.
000002                                     SOURCE
000002       PERFORM P005-VAL-PARM
000002         THRU P005-EXIT.
000002                                     SOURCE
000003     P005-VAL-PARM.
000004     P999-ABEND-PROGRAM.
000004     P999-ABEND-PROGRAM.
000003     P005-EXIT.
000002                                     SOURCE
000002       PERFORM P010-OPEN
000002         THRU P019-EXIT.
000002                                     SOURCE
000003     P010-OPEN.
000003     P019-EXIT.

```

If there are no PERFORMs, or they have already been expanded, the source code for that level displays and the lines tagged as SOURCE (SRC).

Note:

P999-ABEND-PROGRAM is tagged REPEATED because it is referenced more than once at the same level and, conditionally, may be executed more than once.

Removing Source Statements from the Display

To remove displayed levels, follow this step:

- Type ZO in the line command area for PROGRAM-INIT and press Enter. The source code for PROGRAM-INIT is removed from the display, as shown in [Figure 58](#).

Figure 58 • Removing Source Lines from the Display

```

File View Search Options Help
-----
ASG ..... Tree View LINE 1 OF 44
Command ==> ..... Scroll ==> 3

***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001   PROCEDURE DIVISION.
000002     PROGRAM-INIT.
000003       P005-VAL-PARM.
000004         P999-ABEND-PROGRAM.
000004         P999-ABEND-PROGRAM.
000003       P005-EXIT.
000003       P010-OPEN.
000003       P019-EXIT.
000003       P155-CL-SUBTOT.
000004       P159-EXIT.
000003       P120-READ.
000004       P170-FINAL.
000005         P999-ABEND-PROGRAM.
000004       P179-EXIT.
000004       P200-CLOSE.
000005       P129-EXIT.
000004       P129-EXIT.

```

Note:

You can use the ZO line command to remove additional levels from the display. As levels are removed, the plus sign (+) redisplay if lower levels exist.

Switching Between the Editor and Tree View

You cannot modify source code while in the Tree View screen. SmartEdit provides the capability of switching between the Tree View screen and the editor so that you can modify your source code and then continue the tree view.

The JUMP command returns to the editor with the display positioned at the source line corresponding to the cursor position on the Tree View screen. Selecting View ► Return to Tree redisplay the Tree View screen as it was. If the source has changed or another edit session is executed, the Return to Tree action is invalid. You must issue a new Tree function in this case.

To switch to the editor

- 1 Type JUMP in the primary command area, position the cursor at paragraph P005-VAL-PARM, and press Enter.

As shown in [Figure 59](#), the editor screen is positioned at paragraph P005-VAL-PARM after executing the JUMP command.

Figure 59 • Switching to the Editor from a Tree View

```

File View Search Check List Options Help
ASG-SmartEdit - EDIT - VIAINST.CE500001.CNTL(VIAFDEMO COLUMNS 00001 00072
Command ==> Scroll ==> PAGE
027300 027300
027400 P005-VAL-PARM.
027500 027500 IF DBA-DEPT-CODE > 24
027600 027600 PERFORM P999-ABEND-PROGRAM.
027700 027700 IF DBA-DEPT-CODE < 16
027800 027800 PERFORM P999-ABEND-PROGRAM.
027900 027900
028000 028000 P005-EXIT.
028100 028100 EXIT.
028200 028200 SKIP3
028300 028300*****
028400 028400* OPEN ALL FILES *
028500 028500*****
028600 028600
028700 028700 P010-OPEN.
028800 028800 OPEN INPUT MASTERIN.
028900 028900 OPEN OUTPUT MASTER-RPT.
029000 029000 CALL 'DBAOPEN1' USING DBA-DEPT-CODE.
029100 029100 CALL 'DBAOPEN2' USING DBA-DEPT-CODE.
029200 029200

```

Note:

The JUMP command is also available as a line command. To use this feature, type J on the line in the Tree View where you want the editor screen positioned.

- 2 Select View ► Return to Tree and press Enter to return to the Tree View screen. The previous Tree View screen redisplay, as shown in [Figure 60](#). In this example, there were no intervening changes to the source code.

Note:

If you make any changes to the source, the current Tree View becomes invalid. In this case, you must create a new tree view.

Figure 60 • Redisplaying a Program Tree View

```

File View Search Options Help
-----
ASG                                     Tree View                               LINE 1 OF 21
Command ==> _____ Scroll ==> 3

***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001     PROCEDURE DIVISION.
000002     PROGRAM-INIT.
000003         P005-VAL-PARM.
000004             P999-ABEND-PROGRAM.
000004             P999-ABEND-PROGRAM.
000003         P005-EXIT.
000003         P010-OPEN.
000003         P019-EXIT.
000003         P155-CL-SUBTOT.
000004             P159-EXIT.
000003         P120-READ.
000004 +         P170-FINAL.
000004 +         P179-EXIT.
000004 +         P200-CLOSE.
000004 +         P129-EXIT.
000004 +         P120-READ.
000004 +         P129-EXIT.

```

Note:

You can also use the RTV command to return to the tree view screen.

You have now effectively gained a high-level view of the program using the Tree View function. By executing a series of simple steps, you can quickly and easily see a structural representation of your COBOL program.

Using the Intelligent Search within Tree View

In the section ["Using the COBOL Intelligent Search Feature" on page 46](#), you used the SmartEdit Intelligent Search function to locate all references to data items in the program. You can also use this search function in conjunction with the Tree View function to locate modifications and uses for a data item in execution order.

To use the SmartEdit Intelligent Search function

- 1 Select View ► Levels to display all levels of the program structure.
- 2 Select Search ► Data. The Search - Data Name screen, shown in [Figure 61](#), displays.

Figure 61 • Searching for Data Item Usage within a Tree View

```

                                Search - Data Name

Type a data name and select search options. Then press Enter. For a
selection list, enter a pattern (e.g. ABC*) in the name area.

Data Name  LOAN-AMT_

Data References      Indirect Impact      Size change
1  1. All            1  1. None             levels . . . ---
   2. Defs           2. Of Size Change
   3. Uses           3. Of Value Change
   4. Mods

Direction           Options                Action
1  1. All            - No Data Aliasing      1  1. Find
   2. Next           - IN-Clause...          2. Highlight
   3. Previous                          3. Scroll
   4. First                                                  4. Print
   5. Last                                                  5. Punch
                                                6. Exclude
  
```

- 3 Type LOAN-AMT in the Data Name field, select None in the Indirect Impact field, and press Enter.

The results, shown in [Figure 62](#), show that LOAN-AMT is modified in paragraph P120-READ. The next step is to determine how and why.

Figure 62 • Result of Search for Data Item Usage

```

File View Search Options Help
-----
ASG                                     Tree View                               LINE 1 OF 44
Command ==>                               Scroll ==> 3
ASG0639I 8 DATA REFS, 4 DEFS, 2 USES, 2 MODS, FOUND FOR LOAN-AMT.
***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001   PROCEDURE DIVISION.
000002     PROGRAM-INIT.
000003       P005-VAL-PARM.
000004         P999-ABEND-PROGRAM.
000004         P999-ABEND-PROGRAM.
000003       P005-EXIT.
000003       P010-OPEN.
000003       P019-EXIT.
000003       P155-CL-SUBTOT.
000004       P159-EXIT.                                DATA MOD
000003       P120-READ.                                DATA MOD
000004       P170-FINAL.
000005       P999-ABEND-PROGRAM.
000004       P179-EXIT.
000004       P200-CLOSE.                                DATA MOD
000005       P129-EXIT.
000004       P129-EXIT.

```

To view the source code within P120-READ and determine how LOAN-AMT is used and modified, follow this step:

- Type ZI beside the paragraph name P120-READ and press Enter.

The resulting display, shown in [Figure 63](#), indicates that the reference and modification in P120-READ occurs in a READ MASTERIN statement. This is a modification to LOAN-AMT since it is a field within the MASTERIN record.

Figure 63 • Viewing Source Code within a Paragraph

```

File View Search Options Help
-----
ASG                                     Tree View                               LINE 1 OF 56
Command ==>                               Scroll ==> 3
000000 VIAFDEMO.
000001   PROCEDURE DIVISION.
000002     PROGRAM-INIT.
000003       P005-VAL-PARM.
000004         P999-ABEND-PROGRAM.
000004         P999-ABEND-PROGRAM.
000003       P005-EXIT.
000003       P010-OPEN.
000003       P019-EXIT.
000003       P155-CL-SUBTOT.
000004       P159-EXIT.                                DATA MOD
000003       P120-READ.                                SOURCE
000003         IF END-INPUT                                SOURCE
000003           NEXT SENTENCE                                SOURCE
000003         ELSE                                           SOURCE
000003           READ MASTERIN                                SRC/MOD
000003             AT END                                     SOURCE
000003             GO TO P170-FINAL.                            SOURCE
000004       P170-FINAL.

```

Ending Tree View

To return to the SmartEdit editor screen, follow this step:

- ▶ Select File ▶ Exit. The Tree View display is replaced with the SmartEdit editor screen.

Note: _____

You may also press PF3/PF15 to return to the editor screen.

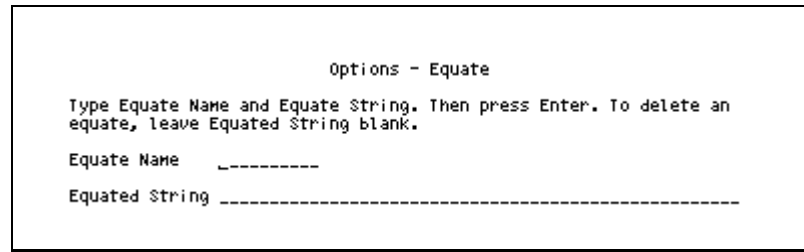
Defining Character Substitution Name

The SmartEdit Equate function provides the capability of defining a substitution name for a character string. You may use it for datanames, concatenated datanames, patterns, etc.

To set up a substitution name

- 1 Select Options ▶ Equate and press Enter. The Options - Equate pop-up, shown in [Figure 64](#), displays.

Figure 64 • Options - Equate Pop-up



```
Options - Equate

Type Equate Name and Equate String. Then press Enter. To delete an
equate, leave Equated String blank.

Equate Name  _
Equated String _____
```

- 2 Type ZIPDATA in the Equate Name field, ZIP-CODE + HLD-ZIP + DET-ZIP-CODE in the Equated String field, and press Enter.

You can then use ZIPDATA with other functions and save the keystrokes of typing in all three datanames each time. The plus sign (+) concatenates the names together.

The List function helps you keep track of all of the Equates you have in an online session. Select List ▶ Equates to display the List - Equates screen ([Figure 109 on page 150](#)). See ["List" on page 147](#) for more information about the List Menu and List pop-ups.

Troubleshooting

When you have problems, SmartEdit provides a robust context-sensitive Help facility to guide you through most situations. However, there are times when Help alone does not answer your usage questions.

This section describes several of the most frequently encountered situations as posed to ASG Customer Support staff.

The CHECK command does not flag the same errors as a batch compile:

Cause	Action
The wrong COBOL version may be specified on the Options - COPY/Include Libraries pop-up (Figure 11 on page 24).	Execute the REFRESH command and select the proper COBOL version.
The wrong compile load library may be specified on the Check - Modules and Libraries pop-up (Figure 106 on page 144).	Select the Check - Modules and Libraries pop-up. Verify with your system administrator that the load libraries specified on this pop-up are correct.

The preprocessor flags errors not normally seen:

Cause	Action
A preprocessor, or the compiler, is flagging I or W level messages when normally only E level or higher are seen.	Select the Check - Compiler Options pop-up (Figure 104 on page 140). The Flag Option field on this screen should be changed to Error.
Unfamiliar DB2 errors are encountered. The preprocessor STDSQL(86) is not normally specified during a batch compile.	Select the Check - Preprocessors pop-up (Figure 105 on page 143). Deselect the STDSQL(86) parameter.

The preprocessor errors are flagged, but COBOL errors are not:

Cause	Action
A preprocessor, or the compiler, is flagging I or W level messages when normally only E level or higher are seen. CHECK is not going on to the compile step.	Select the Check - Compiler Options pop-up. The Flag Option field on this screen should be changed to Error, which allows CHECK to go on to the compile step.

The SmartEdit command does not execute and the program terminates with a COPY MEMBER NOT FOUND message:

Cause	Action
There is a COPY (or ++INCLUDE or -INC etc.) statement in the program for which the copy member cannot be found. The source line containing the COPY error is tagged with an ERR==> tag.	Find the line containing the ERR==> tag. Execute a REFRESH command to get the Options - COPY/Include Libraries screen. Under Copy/Include Data Set Name(s), add a line specifying the dataset containing the copy member. Hit END to return to the edit session.

The SmartEdit command does not execute and the program terminates with DB2 NOT AVAILABLE message:

Cause	Action
SmartEdit has attempted to access the DB2 subsystem and an error has been encountered.	Type HELP to get a message with information about the error. Contact the SmartEdit installer. The VIASBIND job may need to be rerun.

The SmartEdit command does not execute and you get an UNABLE TO RECOVER FROM SYNTAX ERROR NEAR END OF FILE message:

Cause	Action
A syntax error in the program is causing SmartEdit to fail while extracting information about the program. For example, this could be a missing END-EXEC in a DB2 or CICS program.	Execute the CHECK command. If CHECK completes successfully, correct any syntax errors tagged. If CHECK does not complete successfully, run a batch compile of the program and correct any syntax errors tagged. In either case, if no syntax errors were found, call ASG Customer Support.

The SmartEdit command does not execute and you get an INVALID PROGRAM message:

Cause	Action
SmartEdit does not recognize the member being edited as a valid COBOL program. This can be caused by a missing DIVISION statement or an attempt to execute a SmartEdit command on a non-COBOL member, like JCL.	Type HELP to get a message with more information about the error.

You want information about a specific SmartEdit message:

Cause	Action
A message, <i>ASGxxxx</i> , displays on the long message line for which you want a more detailed explanation.	Type <i>HELP XXXX</i> , using the number prefixed from the message. A help screen displays with an explanation of the condition causing the message and suggested actions to take.

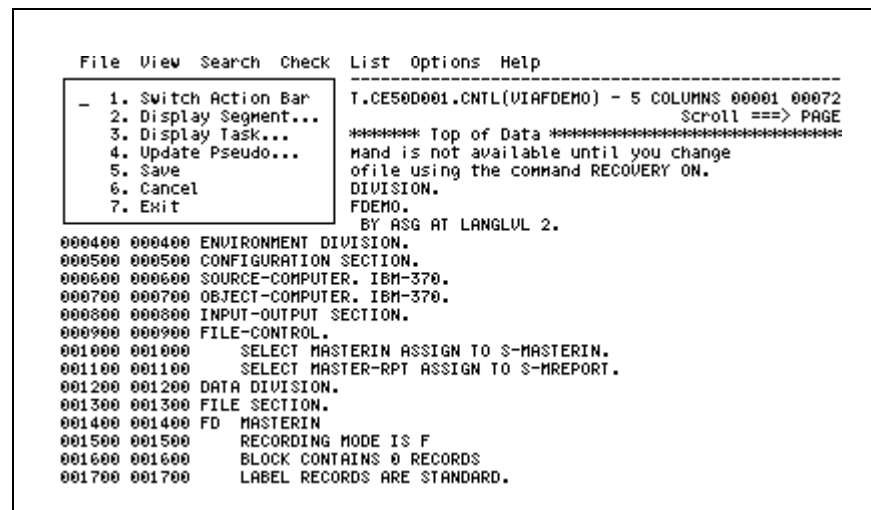
This chapter describes the File pull-down options and contains these sections:

Section	Page
File Pull-down	77
File - Display Logic Segment Entry Pop-up	79
File - Generated Code Update Facility Pop-up	83

File Pull-down

Selecting File on the action bar displays the File pull-down (see [Figure 65](#)). Use this pull-down to save, cancel, or exit a program, and to exit SmartEdit. You can also use the File pull-down to access the display and update functions.

Figure 65 • File Pull-down



If you are using ISPF 4.1 through 4.8, the File pull-down has an additional Switch action bar. The Switch option switches from the SmartEdit action bar to the ISPF Editor action bar. The SWITCH command redisplay the SmartEdit action bar.

Note:

The File pull-down contains different actions when displayed using the Tree view, as shown in [Figure 66 on page 79](#).

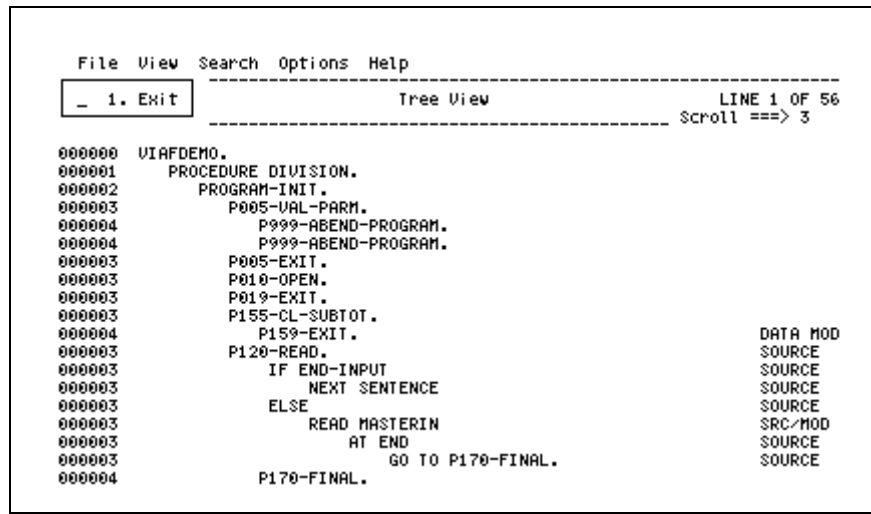
Actions

Action	Description
Display Segment	Displays the File - Display Logic Segment Entry pop-up to display logic segments created in Encore. This action is available only if Encore is installed.
Display Task	Displays the editor screen with results of a Task function performed in Insight. This action is available only if Insight is installed.
Update Pseudo	Displays the File - Generated Code Update Facility pop-up to change pseudo code lines created in SmartTest to actual COBOL source lines. This action is available only if SmartTest in installed.
Save	Saves changes made to the program you are currently editing.
Cancel	Exits the current program without saving changes and without exiting SmartEdit.
Exit	Closes the active program, saves any changes, and redisplay the editor screen. If this action is selected while on Tree View, only the Tree View screen is closed, and the source view is redisplayed.

File Pull-down in Tree View

A different File pull-down displays when you are on the Tree View screen. The only action available is Exit, as shown in [Figure 66](#). If you select File ► Exit, the Tree View screen is closed and the program is redisplayed in Source View.

Figure 66 • File Pull-down in Tree View



File - Display Logic Segment Entry Pop-up

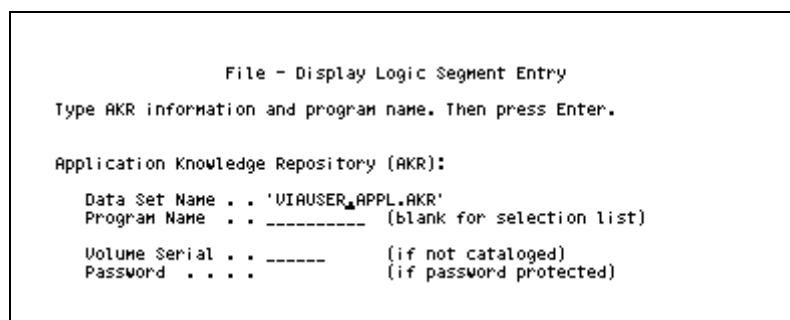
Note:

This pop-up is available only if Encore is installed.

To identify the AKR dataset and program name for the Logic Segment

- Select File ► Display Segment or type DISPLAY. The File - Display Logic Segment Entry pop-up, shown in [Figure 67](#), displays.

Figure 67 • File - Display Logic Segment Entry Pop-up



Fields

Field	Description
Data Set Name	<p>Specifies the AKR dataset name and is a required field. The program name may be typed in this field also. If the TSO ID qualifier is the same as the user ID, type the group-library and type without quotes. For example:</p> <p style="text-align: center;">TEST . AKR</p> <p style="text-align: center;">Or</p> <p style="text-align: center;">TEST . AKR (<i>pgmname</i>)</p> <p>If the TSO ID qualifier is different from the user ID, type the project, group, and type in quotes. For example:</p> <p style="text-align: center;">' TSOID . TEST . AKR '</p> <p style="text-align: center;">Or</p> <p style="text-align: center;">' TSOID . TEST . AKR (<i>pgmname</i>) '</p>
Program Name	<p>Specifies the program name and is an optional field. You can type the program name here or in the Data Set Name field. If you do not enter a program name, the File - Program Name List pop-up displays to select a program name.</p>
Volume Serial	<p>Specifies the volume serial number. This field is required if the dataset is not cataloged. If the dataset is cataloged, this field is optional.</p>
Password	<p>Requires a password if the dataset is password protected.</p>

File - Program Name List Pop-up

Note: _____
 This pop-up is available only if Encore is installed.

Use the File - Program Name List pop-up to select a program name. This pop-up displays when you do not enter a program name on the File - Display Logic Segment Entry pop-up.

The File - Program Name List pop-up, shown in [Figure 68](#), lists all programs containing Logic Segments that exist on the specified AKR, in alphabetical order. You can scroll up and down the list or use the LOCATE command to locate a particular program. After you locate the desired program, use the SELECT command or the S line command to select the program.

Figure 68 • File - Program Name List Pop-up

```

                                File - Program Name List
Command ==> _____ Scroll ==> CSR
Select desired program name. Then press Enter.
AKR: VIAUSER.VIACENxx.AKR

  Name      Alias of    Type      Lines   Date      Time      Jobname
  - - - - -  - - - - -  - - - - -  - - - -  - - - - -  - - - - -
_ VIARDEMO                                     434  DDMMYYYY  HH:MM:SS  SRENDMS2
*****
***** BOTTOM OF DATA *****

```

Fields

Field	Description
AKR	Specifies the complete dataset name of the AKR requested, as entered on the File - Display Logic Segment Entry pop-up.
Line command area	Refers to the area to the left of the Name field, which accepts the S line command. Typing S selects the identified program and displays the File - Select Logic Segment pop-up.
Name	Specifies the name of the programs in the AKR, from the PROGRAM-ID statement. If the analyzed program contains an ENTRY point, Name is the ENTRY point name. If the name in the PROGRAM-ID statement was overridden at the time the analyze job was submitted, Name is the name that was entered in the AKR program name field on the File - Analyze Submit pop-up.
Alias of	Specifies the name of the program that contains the ENTRY point, if the analyzed program contains an ENTRY point. If the name in the PROGRAM-ID statement was overridden at the time the analyze job was submitted, Alias of is the name that was entered in the AKR program name field on the File - Analyze Submit pop-up.
Type	Specifies the type(s) of analysis that was performed on the program. The type of analysis is specified on the File - Analyze Submit pop-up at the time the analyze job is submitted in Encore.
Lines	Specifies the number of lines contained in the program.
Date	Specifies the date the program was analyzed.

Field	Description
Time	Specifies the time the program was analyzed.
Jobname	Specifies the name of the analyze job.

File - Select Logic Segment Pop-up

Note:

This pop is available only if Encore is installed.

After the AKR dataset and program name are identified, the File - Select Logic Segment pop-up, shown in [Figure 69](#), displays to identify the desired Logic Segment name.

Figure 69 • File - Select Logic Segment Pop-up

```

File - Select Logic Segment
Command ==> _____ Scroll ==> CSR

S - Select Logic Segment

Logic Segment Name          Created      Description
- -----
_ DSM-SAVE-001              DD-MMM-YYYY
***** BOTTOM OF DATA *****

```

Fields

Field	Description
Line command area	Refers to the area to the left of the Logic Segment Name field, which accepts the S line command. Typing S selects the Logic Segment for editing.
Logic Segment Name	Lists the names of all the saved Logic Segments in the program. The name is the complete Logic Segment name you entered when the Logic Segment was created.
Created	Indicates the date that the Logic Segment was created. If the Logic Segment was copied from another definition, the date is the COPY date.
Description	Specifies the short description you entered on the File - Save Segment pop-up when the Logic Segment was created. The description may be truncated on the screen due to column limitations.

Displaying Logic Segment

After you have identified the Logic Segment name, the editor screen is redisplayed with all statements in the program that are not contained in the current Logic Segment excluded from the screen, as shown in [Figure 70](#). The logic segment statements are highlighted.

Figure 70 • Displayed Logic Segment

```

File View Search Check List Options Help
-----
ASG-SmartEdit - EDIT - VIAINST.CE50D001.CNTL (VIAFDEMO COLUMNS 00001 00072
Command ==> Scroll ==> CSR
ASG1605I 20 LOGIC SEGMENT LINES WERE NOT FOUND.
- - - - - 470 Line(s) not Displayed
047100 047100
=NOTE= *****
=NOTE= 001400*                               BUILDS NEXT FWD PAGE
=NOTE= 001500      02 RIDF PIC 9(6) .
=NOTE= *****
=NOTE= 001600*
=NOTE= 001700 77 MESSAGES PIC X(39) VALUE ' '.
=NOTE= *****
=NOTE= 001800*
=NOTE= 001900 77 RESPONSE PIC 99(8) COMP.
=NOTE= *****
=NOTE=      02 DFHBHASF PICTURE X VALUE IS '1'.
=NOTE=      02 DFHBHASF PICTURE X VALUE IS 'B'.
=NOTE= *****
=NOTE= 002400*                               FILE RECORD DESCRIPT'N
=NOTE= 002500 01 FILEA.
=NOTE= *****
=NOTE= 000500      02 FILLER REDEFINES MSGF.

```

File - Generated Code Update Facility Pop-up

Note:

This pop-up is available only if SmartTest is installed.

Use the File - Generated Code Update Facility pop-up to enter a dataset and program name to add pseudo code lines (created in SmartTest) to actual COBOL source lines within SmartEdit, making them part of the program.

To display this pop-up, follow this step:

- Select File ► Update Pseudo or type UPDATE. The File - Generated Code Update Facility pop-up, shown in [Figure 71](#), displays.

Figure 71 • File - Generated Code Update Facility Pop-up

```

File - Generated Code Update Facility
Type AKR information and Program Name. Then press Enter.

Application Knowledge Repository (AKR):
Data Set Name . . 'VIAUSEG.APPL.AKR'
Program Name . . _____ (blank for selection list)
Volume Serial . . _____ (if not cataloged)
Password . . . . . (if password protected)

```

Fields

Field	Description
Data Set Name	<p>Specifies the AKR dataset name and is a required field. You can enter the program name in this field also. If the TSO ID qualifier is the same as the user ID, type the group-library and type without quotes. For example:</p> <pre>TEST.AKR</pre> <p>Or</p> <pre>TEST.AKR(<i>pgmname</i>)</pre> <p>If the TSO ID qualifier is different than the user ID, type the project, group, and type in quotes. For example:</p> <pre>'TSOID.TEST.AKR'</pre> <p>Or</p> <pre>'TSOID.TEST.AKR(<i>pgmname</i>)'</pre>
Program Name	<p>Specifies the program name and is an optional field. You can enter the program name here or in the Data Set Name field. If you do not enter a program name, the File - Pseudo Code Program Selection pop-up displays to select a program name.</p>
Volume Serial	<p>Specifies the volume serial number. This field is required if the dataset is not cataloged. If the dataset is cataloged, this field is optional.</p>
Password	<p>Requires a password if dataset is password protected.</p>

File - Pseudo Code Program Selection

Note:

This pop-up is available only if SmartTest is installed.

Use the File - Pseudo Code Program Selection pop-up to select a program. This pop-up displays when you do not type a program name on the File - Generated Code Update Facility pop-up.

The File - Pseudo Code Program Selection pop-up, shown in [Figure 72](#), displays an alphabetical list of all programs in the specified AKR. The list may be scrolled up and down, and you can use the LOCATE command to locate a particular program. Use the SELECT command or the S line command to select the program.

Figure 72 • File - Pseudo Code Program Selection Pop-up

```

File - Pseudo Code Program Selection
Command ==> _____ Scroll ==> 3
Application Knowledge Repository: 'VIAUSER.AMPLAKE'

```

Name	Alias Of	Type	Lines	Date	Time	Jobname
-\$ALLIANCE		INTERNAL	0	27DEC1994	15:38:21	VIAELM2
-\$DCD		INTERNAL	0	16MAY1997	09:50:29	VIAKAP3
-\$METRIC		INTERNAL	1	01APR1994	18:18:59	SRENRAPB
-\$SEGMENTS		INTERNAL	0	12APR1995	10:29:29	SRENJGD
*INTRP21	INTRP2	ST,PLI	3006	15JUL1998	09:20:19	VIAAREBP
##PLANA		TCA PLAN	0	10SEP1999	11:33:06	VIADZR
##PLANB		TCA PLAN	0	10SEP1999	11:36:43	VIADZR
AAAAA		AL	0	30SEP1998	17:06:02	VIAE2KSZ
AAARONS		RC,AL	0	01OCT1998	13:04:08	VIAAREB
ABBEY-AL		RC,AL,TM	0	10FEB1998	05:29:20	VIAARAP
ABBEY-TM		AL,TM	0	24FEB1997	05:41:17	VIAARAP
ABBEYTM		TM	0	02FEB1998	12:22:27	VIAKAS
ABBEY30N		RC	0	18APR1997	14:57:00	VIATTG2
ABBEY300		RC	0	18APR1997	14:56:43	VIATTG2
ALF-NC103		AL	0	25JAN1997	15:32:57	VIALCS4

Fields

Field	Description
Application Knowledge Repository (AKR)	Specifies the complete dataset name of the AKR requested, as entered on the File - Generated Code Update Facility pop-up.
Line command area	Refers to the area to the left of the Name field, which accepts the S line command. Typing S selects the identified program.

Field	Description
Name	Specifies the name of the programs in the AKR, from the PROGRAM-ID statement. If the analyzed program contains an ENTRY point, Name is the ENTRY point name. If the name in the PROGRAM-ID statement was overridden at the time the analyze job was submitted, Name is the name that was entered in the AKR program name field on the File - Analyze Submit pop-up.
Alias of	Specifies the name of the program that contains the ENTRY point if the analyzed program contains an ENTRY point. If the name in the PROGRAM-ID statement was overridden at the time the analyze job was submitted, Alias of is the name that was entered in the AKR program name field on the File - Analyze Submit pop-up.
Type	Specifies the type(s) of analysis that was performed on the program. The type of analysis is specified on the File - Analyze Submit pop-up at the time the analyze job is submitted in SmartTest.
Lines	Specifies the number of lines contained in the program.
Date	Specifies the date the program was analyzed.
Time	Specifies the time the program was analyzed.
Jobname	Specifies the name of the Analyze job.

5

View

This chapter describes the View pull-down options and contains these sections:

Section	Page
View Pull-down	88
View - Tree View Request Pop-up	91
Tree View Screen	93
View - Paragraph Cross Reference Request Pop-up	98
View - Paragraph Cross Reference Pop-up	99
Data Item Offset and Length Pop-up	102
View - Reset Request Pop-up	103
View - Exclude Request Pop-up	104
View - Levels Pop-up	105
View - Scroll Request Pop-up	106

View Pull-down

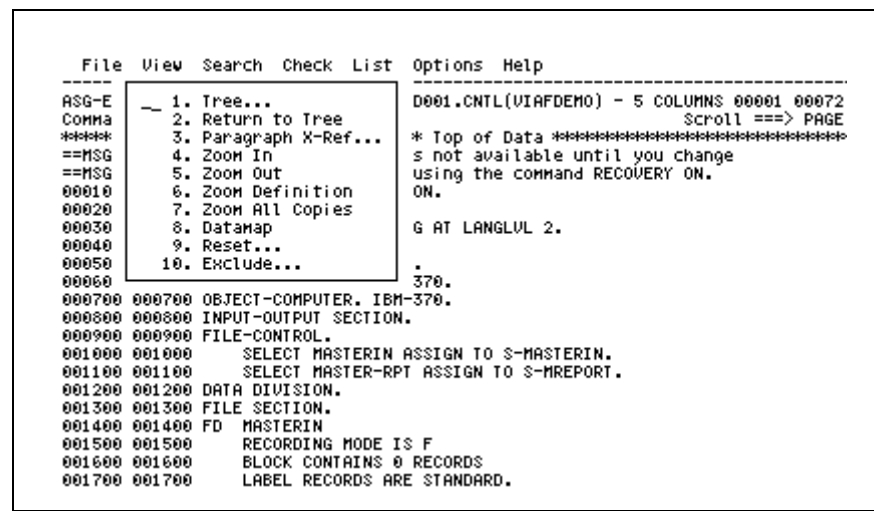
Selecting View on the action bar displays the View pull-down, shown in [Figure 73](#). Use the View pull-down to perform these actions:

- Select the method to view a program
- View transfers of control between paragraphs
- Display or exclude source according to the program hierarchical structure
- Change the level of information viewed
- Display the length and offset of data items contained within a selected data item
- Reset the display
- Exclude source from the display

Note:

The View pull-down contains different actions when displayed in Tree view. See [Figure 74 on page 90](#).

Figure 73 • View Pull-down



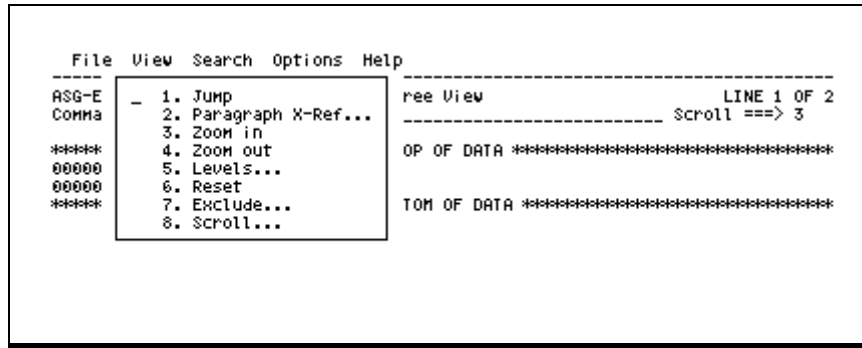
Actions

Action	Description
Tree	Displays the View - Tree View Request pop-up used to select options to customize the Tree View display.
Return to Tree	Redisplays the Tree View screen.
Paragraph X-Ref	Displays the View - Paragraph Cross Reference Request pop-up used to select options for the View - Paragraph Cross Reference pop-up.
	<p>Note:</p> <p>These Zoom-related actions are dependent on cursor location and require use of the ACTION command, which automatically inserts in the command input area, along with a message prompting the user to move the cursor. See "Commands" on page 181 for information about the ACTION command.</p>
Zoom In	Displays source code according to the hierarchical levels of the program. When you select this action, a message displays as a reminder to position the cursor and then press Enter.
Zoom Out	Excludes source code according to the hierarchical levels of the program. When you select this action, a message displays as a reminder to position the cursor and then press Enter.
Zoom Definition	Retrieves the definition of any data items on the line corresponding to the current cursor location, or the first line of the screen.
Zoom All Copies	Retrieves all copy members included by the current source module.
Datamap	Displays the Data Item Offset and Length pop-up showing the length of the data item and the length and offset for any data items that are contained within the selected data item.
Reset	Displays the View - Reset Request pop-up used to select the options to reset.
Exclude	Displays the View - Exclude Request pop-up used to specify the lines to exclude.

View Pull-down in Tree View

[Figure 74](#) shows the View pull-down that displays from the Tree View screen.

Figure 74 • View Pull-down in Tree View



Note:

See ["View - Tree View Request Pop-up" on page 91](#) for information about accessing the Tree View screen.

Actions

Note:

The Jump, Zoom In, and Zoom Out actions are dependent on cursor location and require use of the ACTION command. This command automatically inserts in the command input area, along with a message prompting you to move the cursor. See ["Commands" on page 181](#) for information about the ACTION command.

Action	Description
Jump	Redisplays the editor screen at the location corresponding to the cursor on the Tree View screen. When this action is selected, a message displays as a reminder to position the cursor and then press Enter.
Paragraph X-Ref	Displays the View - Paragraph Cross Reference Request pop-up used to select options for the View - Paragraph Cross Reference pop-up.
Zoom in	Displays source code according to the hierarchical levels of the program. When this action is selected, a message displays as a reminder to position the cursor and then press Enter.
Zoom out	Excludes source code according to the hierarchical levels of the program. When this action is selected, a message displays as a reminder to position the cursor and then press Enter.

Action	Description
Levels	Displays the View - Levels pop-up to specify the new level number.
Reset	Removes highlighting and redisplay excluded lines.
Exclude	Displays the View - Exclude Request pop-up used to specify the lines to exclude.
Scroll	Displays the View - Scroll Request pop-up to specify the option and direction to position the screen.

View - Tree View Request Pop-up

To format the display of the program to be shown on the Tree View screen

- 1 Select View ► Tree. The View - Tree View Request pop-up, shown in [Figure 75](#), displays.

Figure 75 • View - Tree View Request Pop-up

```

View - Tree View Request

Type request options. Then press Enter. For a (name) selection
list, type a pattern (or blanks) in the name field.

Levels 1

Contents                                Range
1  1. Labels                            1  1. Procedure division
   2. Labels and performs                2. All entry points
   3. Labels, performs and GOTOs          3. Perfrange name
   4. Labels, performs, GOTOs and         4. Nested program name
      conditionals

Name -----

Options
/ Repetition

```

- 2 Type the desired configuration parameters and press Enter. The Tree View screen displays.
- 3 Select View ► Levels, or type LEVELS, to change the level depth on the Tree View screen.

Fields

Field	Description
Levels	<p>Specifies the number of hierarchical levels of COBOL source statements to be displayed. The level depth value increases or decreases the amount of detail displayed. The amount of detail increases as the level depth increases. This field is optional.</p> <p>Type the number of levels to be displayed, or type MAX to display the maximum number of levels. The default is 1.</p> <p>When you type MAX, the actual number of levels displayed depends on the Repetition option.</p>
Contents	<p>Specifies the number of the statement type(s) to be displayed. Each statement type is a subset of the COBOL statements contained in the program. SmartEdit classifies COBOL statements into subsets by grouping together COBOL verbs of a similar nature. This field is optional.</p> <p>These are the available subsets:</p> <ul style="list-style-type: none"> • Labels - The PROCEDURE DIVISION statement and all section and paragraph names in the PROCEDURE DIVISION. This is the default. • Performs - Statements containing the PERFORM, SORT, or MERGE verbs. • GOTOs - Statements containing an ALTER or GO TO verb. • Conditionals - Statements or parts of statements that conditionally change the flow of control in a program such as IF, ELSE, and WHEN.
Range	<p>Specifies how much of the program is to be displayed and is an optional field. These are the available ranges:</p> <ul style="list-style-type: none"> • Procedure Division - The entire PROCEDURE DIVISION displays. This is the default. • All Entry Points - The entire PROCEDURE DIVISION and all ENTRY points are displayed. • Perfrange Name - One perform range displays. If you select this option, you must enter the perform range name in the Name field. • Nested Program Name - One nested program displays. If you select this option, you must enter the program name in the Name field.

Field	Description
Name	<p>Specifies the name of the perform range or nested program. This field is required if you selected Range 3 or 4 and is ignored if you selected Range 1 or 2. Type a pattern in this field to display a selection list.</p> <p>If you leave this field blank, a selection list pop-up displays to allow selection of the desired target name. You can enter wildcard characters (? for 1 character; * for zero or more characters) in this field to reduce the size of the selection list.</p> <p>On the selection list pop-up, specify the Name by typing S next to the desired name.</p>
Repetition	<p>Determines whether perform range structures are to be entirely displayed wherever they appear in the program. If not specified, the complete perform range structure displays the first time it appears in the program. Subsequent references to the perform range are not displayed, and are tagged as REPEAT to the right of the PERFORM statement. If specified, the perform range is fully displayed wherever it occurs. The default is to display the perform range only the first time it appears.</p>

Usage Notes

When you request MAX level depth, the actual number of levels displayed may be different, depending on the value in the Repetition field. MAX displays all levels in the program; however, if the maximum level consists of only repeated perform ranges, then the level display is truncated to the next higher level that does not consist exclusively of repeated perform ranges.

Tree View Screen

The Tree View screen shows a textual representation of the execution sequence of a COBOL program. This screen displays when you press Enter on the View - Tree View Request pop-up. The Tree View screen contains a shortened action bar. See ["Usage Notes" on page 97](#) for more information.

The Tree View screen displays logical program units and the relationships between related units. A logical program unit is defined as a PERFORMed range of code including GO TO code, or a CALLED program. Logical program units are displayed as COBOL source lines, and the relationship between each unit is shown by indentation.

When this screen is first displayed, a long message indicates the number of levels displayed, as well as the total number of levels in the program.

To change the number of levels displayed, follow this step:

- Select View ► Levels, or type LEVELS, to display the Tree View screen, shown in [Figure 76](#).

Figure 76 • Tree View Screen

```

File View Search Options Help
-----
ASG-ESU - COBOL Edit           Tree View           LINE 1 OF 21
Command ==>                     Scroll ==> 3

***** ***** TOP OF DATA *****
000000 VIAFDEMO.
000001     PROCEDURE DIVISION.
000002     PROGRAM-BEGIN.
000003         P005-VAL-PARM.
000004             P999-ABEND-PROGRAM.
000004             P999-ABEND-PROGRAM.
000003         P005-EXIT.
000003         P010-OPEN.
000003         P019-EXIT.
000003         P155-CL-SUBTOT.
000004             P159-EXIT.
000003 +       P120-READ.
000003         P129-EXIT.
000002 -       P000-NEXT.
000003         P100-PRINT.
000003 +       P105-NOT-FIRST-TIME.
000003         P110-CONTINUE-PRINT.
000003         P119-EXIT.

```

Tree View Generated Tags

The left column on the Tree View screen indicates the nesting level of each statement. Periods are not shown on the control flow statements generated by Tree View; periods display only on source statements and Tree View generated labels. These tags are placed on lines of special interest to enable greater understanding of program flow:

Tag	Description
+ (plus)	Indicates lower levels of program structure not displayed. To view the lower level structure, use the ZOOMIN command.
: (colon)	Indicates that the statement or label is artificially indented to highlight a conditional statement.
CON	Indicates the control flow verb is part of a conditional (IF, AT END, etc.).
RECUR	Indicates the PERFORM is recursive.
REPEATED REP	If a perform range is repeated, and the REPETITION operand is specified as (or defaulted to) NO, a REPEATED tag is placed on the line of the duplicate perform range.
SOURCE SRC	Specifies that the displayed line is actual source code.

The most basic Tree View diagram displays paragraph and section labels. All executed labels from a PERFORM statement are indented under the paragraph label containing the PERFORM. Program exits are also displayed.

Control flow statements can also be displayed, enabling greater understanding of how the labels are executed. Indentation is directly related to the nesting level of the PERFORM statement.

Primary Commands

These primary commands perform the specified functions on the Tree View display when entered on the command line prompt:

Command	Description
BRANCH	Locates the occurrence of the PERFORM within the Tree View display. If the label is not shown, a message displays.
END	Returns to the editor screen at the location where the TREEVIEW command or the RETURN TO TREE command is entered.
JUMP	Returns to the editor screen at the location corresponding to the line where this command is issued on the Tree View screen. Jump also functions as the J line command.
LEVELS <i>n</i>	Redisplays the Tree View screen to the <i>n</i> th level. If there are levels less than <i>n</i> , they expand to the requested level. If there are levels greater than <i>n</i> , they are removed. The MAX operand displays the maximum nesting level. However, if the maximum nesting level consists of only repeated PERFORM ranges, then the nested level is truncated to the next higher level that does not consist exclusively of repeated PERFORM ranges. The LEVELS command works globally throughout the displayed program. The default level is 1.
RESET	Removes highlighting, erases tags in columns 73 through 80, and redisplays excluded lines.

Command	Description
ZOOMIN	<p>Expands the Tree View diagram to the next lower hierarchical level. Typing ZOOMIN on a label expands all PERFORMs contained by the paragraph. If there are no more hierarchical levels, or they have already been expanded, the source displays for that label.</p> <p>If all levels have been displayed, and you type ZOOMIN, an error message displays. You cannot type ZOOMIN on a source line.</p>
ZOOMOUT	<p>Removes levels. The first level removed is the source. If there is no source displayed, then lower levels are removed. If both source and lower levels have been removed, then no more levels are removed.</p>

Paragraphs or PERFORMs are marked as containing results (i.e., they are tagged as USE, MOD, or REF). If the results at the source level are desired, ZOOMIN may be used to see the source and the actual results. Results are not displayed if they are not within Tree View's bounds. The RECALL and REPEAT commands work the same way in Tree View as they do in the editor.

The EXCLUDE, FINDXTND, and HIGH commands function the same way in Tree View as they do in the editor except these operands are not applicable in Tree View:

```
FIRST
LAST
PREV
NEXT
```

You can execute the LPRINT and LPUNCH commands in the Tree View screen, however, the only valid operands are * and NOTE. LPRINT and LPUNCH copy the specified target to the List and Punch file, respectively.

You can execute the SCROLL command on the Tree View screen, however, only these operands are valid in Tree View:

```
HI
FIRST
LAST
NHI
NEXT
PREV
```

Note: _____
 These operands are not valid with datanames in Tree View.

Line Commands

These line commands perform the specified functions on the Tree View screen.

Command	Function
Fn	Redisplays the specified number of excluded lines beginning with the first excluded line. The default is 1.
J	Returns to the editor screen at the location corresponding to the line where this command is issued on the Tree View screen. This command also functions as the JUMP primary command.
Ln	Redisplays excluded lines starting with the last line in the block of excluded lines. The default is 1.
Sn	Redisplays excluded lines starting with the first line in the block of excluded lines. The default is 1.
SS	Redisplays a block of lines. Type on the first and last line in the block of lines to be redisplayed.
Xn	Excludes the specified number of lines from being displayed. The default is 1.
XX	Excludes a block of lines from being displayed. Type this command on the first and last line to excluded.
ZI	Performs the ZOOMIN command at the line on that it is entered. See the ZOOMIN primary command in "Commands" on page 181 , for more information.
ZO	Performs the ZOOMOUT command at the line on that it is entered. See the ZOOMOUT primary command in "Commands" on page 181 , for more information.

Usage Notes

The Tree View screen contains a shortened action bar, including these actions: File, View, Search, Options, and Help. The Check and List actions are not available in Tree View.

View - Paragraph Cross Reference Request Pop-up

To set options for the View - Paragraph Cross Reference pop-up, follow this step:

- Select View ► Paragraph X-Ref. The View - Paragraph Cross Reference Request pop-up, shown in [Figure 77](#), displays.

Figure 77 • View - Paragraph Cross Reference Request Pop-up

```

View - Paragraph Cross Reference Request

Type Target Name(s) and select a Direction. Then press Enter.
For a selection list, type a pattern in the Target field (except
for Line Range).

Target                                     Direction
Use cursor -                               1  1. Previous
Program . . .                             2. Next
Subset . . .
Label . . .
Line Range . . .
  
```

Fields

Field	Description
Target	Specifies the fields that indicate the basis for the Paragraph Cross Reference. You can enter a pattern with wildcard characters (? for 1 character; * for zero or more characters) in any of the fields, except Use cursor and Line Range, to display a selection list pop-up.
Use cursor	Specifies the target by positioning the cursor on the desired target. When this Target type is selected, the editor screen is redisplayed to allow cursor placement.
Program	Displays all paragraph names containing the program name you specify.
Subset	Displays all paragraph names containing statements of the specified subset type. See "Product Overview" on page 5 for a description of each subset.
Label	Displays all paragraph names containing the specified label. Type any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.

Field	Description
Line Range	Displays all paragraph names containing the specified line(s). Type a single line number or range of lines. Line numbers are displayed in columns 1 through 6.
Direction	<p>Specifies the direction of the transfer.</p> <p>Previous. Generates a list of paragraphs that transfer control to the target. This is the default.</p> <p>Next. Generates a list of paragraphs to which the target transfers control.</p>

View - Paragraph Cross Reference Pop-up

To designate how control is transferred to or from the target paragraphs, follow this step:

- Press Enter on the View - Paragraph Cross Reference Request pop-up. The View - Paragraph Cross Reference pop-up, shown in [Figure 78](#), displays.

You can also use the LPRINT * command on this pop-up to copy the data to the List file.

Figure 78 • View - Paragraph Cross Reference Pop-up

```

View - Paragraph Cross Reference
Command ==> _----- Scroll ==> 3
Target(s): VIAFDEMO
Direction: Previous

  A : Add to target      P : Execute PREF      S : Select for viewing
-----
  Comes From           How           Target Paragraph(s)
-----
- PROCEDURE DIVISION    PROGRAM ENTRY  PROCEDURE DIVISION
- " "                   PREV PARA     "
- " "                   PERF,PREV     PROGRAM-INIT
- PROGRAM-INIT          PERFORM       P000-NEXT
- P000-NEXT             PREV PARA     "
- PROGRAM-INIT          PREV PARA     P000-EXIT
- P000-EXIT             PERFORM       P005-VAL-PARM
- P005-VAL-PARM         PREV PARA     "
- PROGRAM-INIT          PREV PARA     P005-EXIT
- P005-EXIT             PERFORM       P010-OPEN
- P010-OPEN             PREV PARA     "
- P010-OPEN             PERFORM       P019-EXIT

```

Fields

Field	Description
Target(s)	Specifies the paragraph name(s) of the requested target.
Direction	Indicates the transfer direction. Previous highlights paragraphs from which control is transferred. Next highlights paragraphs to which control is transferred. The default is Previous.
Line command area	<p>Refers to the area to the left of the Comes From or Target Paragraph(s) field, which accepts these line commands:</p> <ul style="list-style-type: none">• A - Add the selected paragraph to the target paragraph(s). If the direction is Previous, the target paragraph is added at the top of the list. If the direction is Next, the target paragraph is added to the end of the list.• P - Execute another paragraph cross-reference.• S - Return to the source code with the first line of the paragraph displayed at the top of the screen.
Comes From	Displays when the direction is Previous. It lists the paragraph names that transfer control to the target paragraph(s). All paragraphs that relate to the target paragraph are grouped together.
Goes To	Displays when the direction is Next. It lists the paragraph names that receive control from the target paragraph(s).

Field	Description																										
How	<p>Indicates the way control is transferred from or to the target paragraph.</p> <p>If the target is reached through more than one occurrence of the same transfer method, the number of occurrences, in parentheses, follows the method of transfer. For example:</p> <p>PERF(2), GO(3) means that the target was reached through 2 PERFORMs and 3 GO TOs.</p> <p>You can transfer control using these values:</p> <table> <tr> <th>Value</th><th>Abbreviation</th></tr> <tr> <td>PERFORM</td><td>PERF(3),PERF,P(2),P</td></tr> <tr> <td>GOTO</td><td>GO(3), G(2), G</td></tr> <tr> <td>ALTERED GOTO</td><td>ALTER, AG</td></tr> <tr> <td>PREV PARA</td><td>PREV</td></tr> <tr> <td>NEXT PARA</td><td>NEXT</td></tr> <tr> <td>GOTO DEPENDING</td><td>GODEP, GD</td></tr> <tr> <td>FALLTHRU</td><td>FALL, F</td></tr> <tr> <td>USE ON ERROR</td><td>USE ERROR, U</td></tr> <tr> <td>USE FOR DEBUG</td><td>USE DEBUG, U</td></tr> <tr> <td>USE BEFORE REPORT</td><td>USE REPORT, U</td></tr> <tr> <td>*CALL INTERNAL</td><td>CLL INT, CI</td></tr> <tr> <td>*CALL RETURN</td><td>CLL RTN, CR</td></tr> </table> <p>* COBOL II Release 3 and later programs containing subprograms.</p>	Value	Abbreviation	PERFORM	PERF(3),PERF,P(2),P	GOTO	GO(3), G(2), G	ALTERED GOTO	ALTER, AG	PREV PARA	PREV	NEXT PARA	NEXT	GOTO DEPENDING	GODEP, GD	FALLTHRU	FALL, F	USE ON ERROR	USE ERROR, U	USE FOR DEBUG	USE DEBUG, U	USE BEFORE REPORT	USE REPORT, U	*CALL INTERNAL	CLL INT, CI	*CALL RETURN	CLL RTN, CR
Value	Abbreviation																										
PERFORM	PERF(3),PERF,P(2),P																										
GOTO	GO(3), G(2), G																										
ALTERED GOTO	ALTER, AG																										
PREV PARA	PREV																										
NEXT PARA	NEXT																										
GOTO DEPENDING	GODEP, GD																										
FALLTHRU	FALL, F																										
USE ON ERROR	USE ERROR, U																										
USE FOR DEBUG	USE DEBUG, U																										
USE BEFORE REPORT	USE REPORT, U																										
*CALL INTERNAL	CLL INT, CI																										
*CALL RETURN	CLL RTN, CR																										
Target Paragraph(s)	Specifies the paragraphs requested as the target in the PREF command. Ditto marks (") are shown when multiple paragraphs transfer control to the same paragraph.																										

Data Item Offset and Length Pop-up

To display the length of the data item and the length and offset for any data items that are contained within the selected data item, follow this step:

- Select View ► Datamap. The Data Item Offset and Length pop-up, shown in [Figure 79](#), displays.

Figure 79 • Data Item Offset and Length Pop-up

Offset	Contains Data Items	Length
0	05 DET-CC	1
1	05 FILLER	17
18	05 DET-NUMBER	6
24	05 FILLER	49
73	05 FILLER	16
89	05 DET-LOAN-AMT	18
107	05 FILLER	2
109	05 FILLER	17
126	05 DET-START-DATE	6
132	05 FILLER	3

***** BOTTOM OF DATA *****

Fields

Field	Description
Data Item	Displays the name of the selected data item.
Length	Displays the total length of all data items contained within the selected data item.
Offset	Displays the number of characters that the data item is offset in the selected data item.
Contains Data Items	Displays all of the data item(s) that the selected data item contains.
Length	Displays the length of each data item that the selected data item contains.

View - Reset Request Pop-up

To remove highlighting and tags, and to redisplay excluded lines:

- Select View ► Reset. The View - Reset Request pop-up, shown in [Figure 80](#), displays.

Figure 80 • View - Reset Request Pop-up



Fields

Field	Description
All	Optional. If specified, all conditions indicated by each of the other fields on this pop-up are reset. If not specified, only those options selected are reset. The default is to reset all conditions.
Highlights	Optional. If specified, any line that has been highlighted returns to nonhighlighted. If not specified, highlighted lines remain highlighted unless the All option is specified. The default is lines remain highlighted.
Tags	Optional. If specified, tags in columns 73 through 80 are erased. If not specified, tags remain unless the All option is specified. The default is lines remain tagged.
Excluded Lines	Optional. If specified, excluded lines redisplay. If not specified, excluded lines remain excluded unless the All option is specified. The default is lines remain excluded.
Zoom Copies	Optional. If specified, copy members included in the active source module are refreshed. If not specified, copy members are not brought in unless the All option is specified. The default is copy members are not refreshed.

View - Exclude Request Pop-up

Use the View - Exclude Request pop-up to specify lines to be excluded from the display. Excluded lines are represented by a line of dashes and text stating *n* lines not displayed.

To specify lines to be excluded, follow this step:

- Select View ► Exclude. The View - Exclude Request pop-up, shown in [Figure 81](#), displays.

Figure 81 • View - Exclude Request Pop-up

```

View - Exclude Request

To exclude specific lines from the program being
viewed, select the desired option. Then press
Enter. For the "string" option, type the string
to be excluded.

Options
1. All lines
2. All highlighted lines
3. All non-highlighted lines
4. All lines containing a string

String -----

```

Fields

Field	Description
Options	<p>Specifies which lines to exclude from the screen display. The default is 1, All lines. These are the valid values:</p> <ul style="list-style-type: none"> 1. All lines. Excludes all lines from the screen display. This is the default. 2. All highlighted lines. Excludes all lines that have been highlighted as the result of one or more actions or commands from the screen display. 3. All non-highlighted lines. Excludes all non-highlighted lines from the screen display. 4. All lines containing a string. Excludes all lines containing the string specified in the String field. If this Option is selected, the String field is required.
String	<p>Specifies a string of alphanumeric or DBCS characters, This field is required if you selected Option 4 and is ignored if you select any other Option.</p>

View - Levels Pop-up

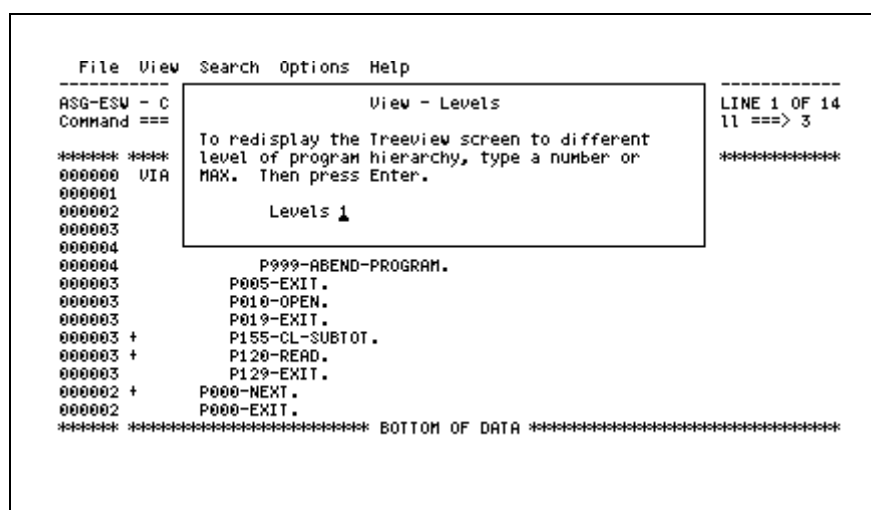
Note:

The View - Levels pop-up is available only in the Tree View facility.

To redisplay the current Tree View screen to the specified level, follow this step:

- Select View ► Levels in Tree View. The View - Levels pop-up, shown in [Figure 82](#), displays.

Figure 82 • View - Levels Pop-up



Field

Field	Description
Levels	Optional. Specify the number of hierarchical levels to be displayed. Change the level depth value to increase or decrease the amount of detail displayed. The amount of detail increases as the level depth increases. Type the number of levels to be displayed, or type the value MAX to display the maximum number of levels. The default is 1.

View - Scroll Request Pop-up

Note:

The View - Scroll Request pop-up is available only in the Tree View facility.

To display a highlighted or non-highlighted line within Tree View, follow this step:

- Select View ► Scroll. The View - Scroll Request pop-up, shown in [Figure 83](#), displays.

Figure 83 • View - Scroll Request Pop-up

```

Menu RefList RefMode Utilities LMF Worksta
-----
Edit Entry Panel
Command ==>

ISPF Library:
Project . . . UIAINXX
Group . . . CE70TEST
Type . . . CNTL
Member . . . (Blank or patter

Other Partitioned, Sequential or VSAM Data Set:
Data Set Name . . .
Volume Serial . . . (If not cataloge

Workstation File:
File Name . . .
  
```

Fields

Field	Description
Scroll Options	<p>Optional. Type the number corresponding to the type of line desired. The default is 1, Highlighted Line.</p> <ol style="list-style-type: none"> 1. Highlighted Line. Scrolls to the highlighted line specified in the Scroll Direction field. 2. Non-Highlighted Line. Scrolls to the non-highlighted line specified in the Scroll Direction field.
Scroll Direction	<p>Optional. Type the number corresponding to the direction desired. The default is 1, First.</p> <ol style="list-style-type: none"> 1. First. Scrolls to the first occurrence of the line specified in the Scroll Options field. 2. Last. Scrolls to the last occurrence of the line specified in the Scroll Options field. 3. Prev. Scrolls to the previous occurrence of the line specified in the Scroll Options field. 4. Next. Scrolls to the next occurrence of the line specified in the Scroll Options field.

6

Search

This chapter describes the Search pull-down options and contains these sections:

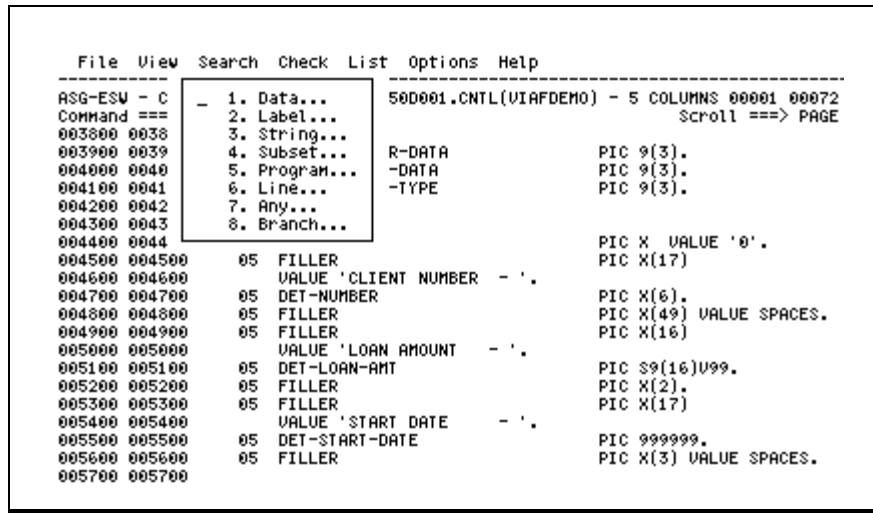
Section	Page
<u>Search Pull-down</u>	<u>108</u>
<u>Search - Data Name Pop-up</u>	<u>109</u>
<u>Search - Label Name Pop-up</u>	<u>113</u>
<u>Search - Pattern String Pop-up</u>	<u>116</u>
<u>Search - COBOL Subset Name Pop-up</u>	<u>120</u>
<u>Search - Program Name Pop-up</u>	<u>123</u>
<u>Search - Line Range Pop-up</u>	<u>126</u>
<u>Search - Any/Unknown Type Pop-up</u>	<u>129</u>
<u>Search - Branch Request Pop-up</u>	<u>132</u>
<u>Selection List Pop-ups</u>	<u>134</u>
<u>Search - IN Clause Pop-up</u>	<u>135</u>

Search Pull-down

To conduct a COBOL intelligent search of the source code for one or all occurrences of a specified target, follow this step:

- Select Search on the action bar to display the Search pull-down, shown in [Figure 84](#).

Figure 84 • Search Pull-down



Actions

Action	Description
Data	Displays the Search - Data Name pop-up used to search for data items.
Label	Displays the Search - Label Name pop-up used to search for a specified label name.
String	Displays the Search - Pattern String pop-up used to search for a character string.
Subset	Displays the Search - COBOL Subset name pop-up used to search for statements corresponding to various COBOL subsets.
Program	Displays the Search - Program Name pop-up used to search for statements in subprograms.
Line	Displays the Search - Line Range pop-up used to search for statements within line ranges.

Action	Description
Any	Displays the Search - Any/Unknown Type pop-up used to search for targets of unknown types.
Branch	Displays the Search - Branch Request pop-up used to branch to a specific location.

Search - Data Name Pop-up

To find, highlight, scroll, print, punch, or exclude datanames, follow this step:

- Select Search ► Data. The Search - Data Name pop-up, shown in [Figure 85](#), displays.

Figure 85 • Search - Data Name Pop-up

Search - Data Name

Type a data name and select search options. Then press Enter. For a selection list, enter a pattern (e.g. ABC*) in the name area.

Data Name _____

Data References 1 1. All 2. Defs 3. Uses 4. Mods	Indirect Impact 1 1. None 2. Of Size Change 3. Of Value Change	Size change levels . . . ____
---	--	---

Direction 1 1. All 2. Next 3. Previous 4. First 5. Last	Options - No Data Aliasing - IN-Clause...	Action 1 1. Find 2. Highlight 3. Scroll 4. Print 5. Punch 6. Exclude
---	--	---

Note:

The Search - Data Name pop-up is different when displayed in Tree view. See [Figure 86 on page 113](#).

Fields

Field	Description
Data Name	<p>Specifies a COBOL dataname or qualified COBOL dataname. Dataname refers to any valid COBOL reference for a data element.</p> <p>Datanames may be concatenated by placing a plus sign (+) between the datanames.</p> <p>To display the Selection List pop-up, leave the Data Name field blank, or type a pattern with wildcard characters (? for one character; * for zero or more characters). See "Selection List Pop-ups" on page 134 for more information.</p>
Data References	<p>Specifies the type of dataname reference. This restricts the search to only those datanames that are defined, used, or modified. These are the valid data references:</p> <ul style="list-style-type: none"> • All - Includes occurrences of the dataname where its value is defined, tested, used, set, or modified. This is the default. • Defs - Includes definitions of the dataname in the DATA DIVISION. • Uses - Includes occurrences of the dataname where its value is being tested or used. • Mods - Includes occurrences of the dataname where its value is being set or modified.
Indirect Impact	<p>Restricts the search to include only occurrences of datanames that are indirectly affected by the specified dataname. These are the valid restrictions:</p> <ul style="list-style-type: none"> • None - Includes occurrences of the dataname (and aliases if specified) where it is directly tested, used, set, modified, or defined (based on the value of the Data References field). This is the default. • Of Size Change - Includes datanames that could be directly or indirectly affected if the size of the specified dataname were to be changed. • Of Value Change - Includes datanames that could be directly or indirectly affected if the value of the specified dataname were to be changed.
Size change levels	<p>Specifies a number that identifies the depth of the indirect references. When you use this field, all references that are affected within the specified level are shown. If left blank, all levels are included.</p>

Field	Description
Direction	<p>Specifies a direction to restrict the search to a specific direction or occurrence. These are the valid directions:</p> <ul style="list-style-type: none"> • All - Searches for all occurrences of the dataname and displays the number found in the short/long message field. This is the default. • Next - Searches forward from the cursor position to the next occurrence of the dataname. • Previous - Searches backward from the cursor position to the previous occurrence of the dataname. • First - Searches from the top of the source file to the first occurrence of the dataname. • Last - Searches backward from the bottom of the source file to the last occurrence of the dataname. <p>Note: _____ Direction is not available in Tree View.</p>
No Data Aliasing	<p>Specifies that aliases of the dataname will be ignored. These are the valid aliases:</p> <ul style="list-style-type: none"> • Parent - higher level group item. • Child - lower level item. • Rename/Redefinition - renamed, redefined, or 88 level items. • Blank - aliases of the dataname are included. This is the default.

Field	Description
IN-Clause	Displays the Search - IN Clause pop-up used to restrict the source lines considered for a search. If left blank, the entire source code program is searched. The default is the entire program.
Action	<p>Specifies the action to be performed on the dataname. The default is Find. These are the actions you can perform:</p> <ul style="list-style-type: none">• Find - Searches for one or all occurrences of the dataname. You can specify DBCS identifiers or patterns to locate DBCS strings.• Highlight - Highlights source code lines containing the dataname. Lines that are already highlighted are not reset.• Scroll - Positions the screen to the first line containing the dataname. Highlighted lines remain unchanged. This action is available only on the editor screen. To scroll in Tree View, select View ► Scroll.• Print - Copies lines containing the dataname to the List file. This action is available only on the editor screen. The List file is processed on the Options - Log/List/Punch Definition pop-up.• Punch - Copies lines containing the dataname to the Punch file. This action is available only on the editor screen. The Punch file is processed on the Options - Log/List/Punch Definition pop-up.• Exclude - Omits source code lines containing occurrences of the dataname from the display. <p>Note: _____ Scroll, Print, and Punch are not available in Tree View. _____</p>

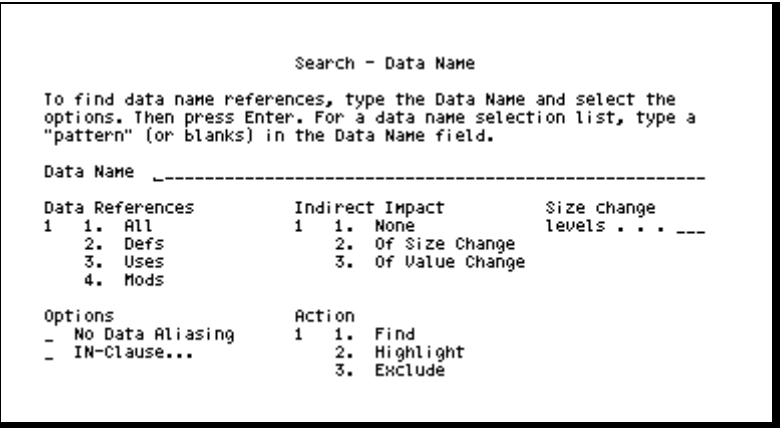
Usage Notes

To find, highlight, or exclude datanames in Tree view, follow this step:

- Select View ► Scroll. The Search - Data Name pop-up, shown in [Figure 86](#), displays.

The Scroll, Print, and Punch actions are not available in Tree View. Also, Direction is not valid in Tree View.

Figure 86 • Search - Data Name Pop-up in Tree View

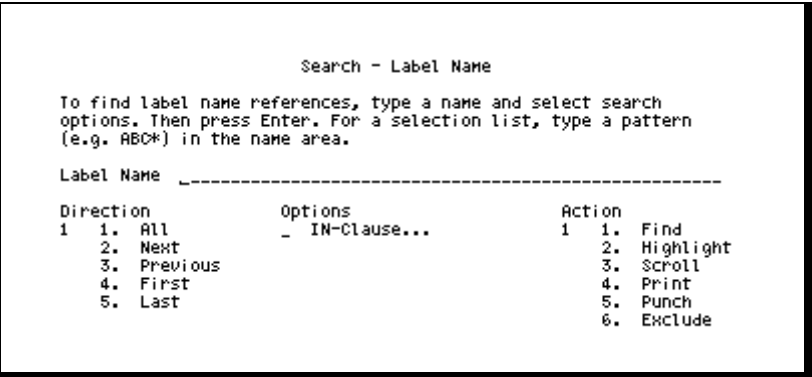


Search - Label Name Pop-up

To find, highlight, scroll, print, punch, or exclude a specified label name, follow this step:

- Select Search ► Label. The Search - Label Name pop-up, shown in [Figure 87](#), displays.

Figure 87 • Search - Label Name Pop-up



Note: _____

The Search - Label Name pop-up is different when displayed in Tree View.

Fields

Field	Description
Label Name	<p>Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.</p> <p>To display the Selection List pop-up, leave the Label Name field blank, or type a pattern with wildcard characters (? for one character; * for zero or more characters). See "Selection List Pop-ups" on page 134.</p>
Direction	<p>Specifies a direction to restrict the search to a specific direction or occurrence. The default is All. Direction is not available in Tree View.</p> <ul style="list-style-type: none">• Next - Searches forward from the cursor position to the next occurrence of the label name.• Previous - Searches backward from the cursor position to the previous occurrence of the label name.• First - Searches from the top of the source file to the first occurrence of the label name.• Last - Searches backward from the bottom of the source file to the last occurrence of the label name.• All - Searches for all occurrences of the label name and displays the number found in the short/long message field. This is the default.

Field	Description
Options	Displays the Search - IN Clause pop-up used to restrict the source lines to be considered for a search. If left blank, the entire source code program is searched. The default is the entire program.
Action	<p>Specifies the action to be performed on the label name. The default is Find. Scroll, Print, and Punch are not available in Tree View. These are the valid actions:</p> <ul style="list-style-type: none"> • Find - Searches for one or all occurrences of the label name. You can specify DBCS identifiers or patterns to locate DBCS strings. • Highlight - Highlights source code lines containing the dataname. Lines that are already highlighted are not reset. • Scroll - Positions the screen to the first line containing the label name. Highlighted lines remain unchanged. This action is available only on the editor Screen. To scroll in Tree View, select View ► Scroll. • Print - Copies lines containing the label name to the List file. This action is available only on the editor screen. The List file is processed on the Options - Log/List/Punch Definition pop-up. • Punch - Copies lines containing the label name to the Punch file. This action is available only on the editor screen. The Punch file is processed on the Options - Log/List/Punch Definition pop-up. • Exclude - Omits lines containing occurrences of the label name from the display.

Usage Notes

To find, highlight, or exclude label names in Tree view, follow this step:

- Select View ► Scroll. The Search - Label Name pop-up in Tree view, shown in [Figure 88](#), displays.

Figure 88 • Search - Label Name Pop-up in Tree View

```

Search - Label Name

To find label name references, type the Label Name and select the
options. Then press Enter. For a Label Name selection list, type a
pattern (or blanks) in the Label Name field.

Label Name _____

Options          Action
- IN-Clause...   1  1. Find
                  2. Highlight
                  3. Exclude
  
```

Note:

The Direction, Scroll, Print, and Punch actions are not available in Tree View.

Search - Pattern String Pop-up

To find, highlight, scroll, print, punch, or exclude a specified pattern string, follow this step:

- Select Search ► String. The Search - Pattern String pop-up, shown in [Figure 89](#), displays.

Figure 89 • Search - Pattern String Pop-up

```

Search - Pattern String

Type a string and select search options. Then press Enter.

String _____

String type      Location
1  1. Simple      1  1. Any
    2. Hexadecimal 2. Word
    3. Text        3. Prefix
    4. Picture      4. Suffix

Direction      Options          Action
1  1. All         - IN-clause...   1  1. Find
    2. Next                               2. Highlight
    3. Previous                               3. Scroll
    4. First                                           4. Print
    5. Last                                           5. Punch
                                                    6. Exclude

Begin column ___ End column ___
  
```

The Search - Pattern String pop-up is different when displayed in Tree View. See [Figure 90 on page 119](#).

Fields

Field	Description
String	Specifies a string of alphanumeric or DBCS characters. It is not necessary to enclose the string in single or double quotes. All characters you enter are used for the pattern search.
String type	<p>Specifies the type of pattern string. The default is Simple.</p> <ul style="list-style-type: none"> Simple - Indicates a string that is not a hexadecimal, text, or a picture string. Hexadecimal - Specific unprintable characters may be specified by giving the EBCDIC hexadecimal value. Text - A character string may be entered regardless of upper or lowercase by using the text option. Picture - A string profile may be entered instead of exact characters. Nine special characters are defined by SmartEdit for use in picture strings. These special characters may be combined with other characters. These are the special characters: <ul style="list-style-type: none"> P'=' Any character P'¬' Any nonblank character P'.' Any nondisplay character P'#' Any numeric character P'-' Any non-numeric character P'@' Any alphabetic character (upper or lowercase) P'<' Any lowercase alphabetic character P'>' Any uppercase alphabetic character P'\$' Any special character (not alphabetic or numeric)
Location	<p>Specifies the location to search for the pattern string. The default is Any.</p> <ul style="list-style-type: none"> Any - A string anywhere in the source code. Word - A string preceded and followed by any non-alphanumeric character (except hyphen). Prefix - A word that begins with the specified string. Suffix - A word that ends with the specified string.

Field	Description
Options	Displays the Search - IN Clause pop-up used to restrict the source lines to be considered for a search. If left blank, the entire source code program is searched. The default is the entire program. See "Search - IN Clause Pop-up" on page 135 for more information.
Direction	<p>Specifies a direction to restrict the search to a specific direction or occurrence. Direction is not available in Tree View.</p> <ul style="list-style-type: none"> • All - Searches for all occurrences of the string and displays the number found in the short/long message field. This is the default. • Next - Searches forward from the cursor position to the next occurrence of the string. • Previous - Searches backward from the cursor position to the previous occurrence of the string. • First - Searches from the top of the source file to the first occurrence of the string. • Last - Searches backward from the bottom of the source file to the last occurrence of the string.
Action	<p>Specifies the action to be performed on the string. The default is Find. Scroll, Print, and Punch are not available in Tree View.</p> <ul style="list-style-type: none"> • Find - Searches for one or all occurrences of the string. You can specify DBCS identifiers or patterns to locate DBCS strings. • Highlight - Highlights source code lines containing the string. Lines that are already highlighted are not reset. • Scroll - Positions the screen to the first line containing the string. Highlighted lines remain unchanged. This action is available only on the editor screen. To scroll in Tree View, select View ► Scroll. • Print - Copies lines containing the string to the List file. This action is available only on the editor screen. The List file is processed on the Options - Log/List/Punch Definition pop-up. • Punch - Copies lines containing the string to the Punch file. This action is available only on the editor screen. The Punch file is processed on the Options - Log/List/Punch Definition pop-up. • Exclude - Omits lines containing occurrences of the string from the display.

Field	Description
Begin column	Specifies the column number where the search is to begin. The default is column 7. Begin columns is not available in Tree View.
End column	Specifies the column number where the search is to end. The default is column 66. End column is not available in Tree View.

Usage Notes

To find, highlight, or exclude lines containing a string while in Tree view, follow this step:

- Select View ► Scroll. The Search - Pattern String pop-up, shown in [Figure 90](#), displays.

Note:

The Direction, Scroll, Print, and Punch actions are not available in Tree view.

Figure 90 • Search - Pattern String Pop-up in Tree View

```

Search - Pattern String

To find statements containing a "character string", type the string
and select the options. Then press Enter.

String _-----

String Type          Location          Action
1  1. Simple          1  1. Any          1  1. Find
   2. Hexadecimal     2  2. Word         2  2. Highlight
   3. Text             3  3. Prefix        3  3. Exclude
   4. Picture          4  4. Suffix
Options
_ IN-Clause...

```

Search - COBOL Subset Name Pop-up

To find, highlight, scroll, print, punch, or exclude statements corresponding to various COBOL subsets, follow this step:

- Select Search ► Subset. The Search - COBOL Subset Name pop-up, shown in [Figure 91](#), displays.

Note:

The Search - COBOL Subset Name pop-up is different when displayed in Tree View. See [Figure 92 on page 123](#) for more information.

Figure 91 • Search - COBOL Subset Name Pop-up

Search - COBOL Subset Name

Type a subset name and select search options. Then press Enter. For a selection list, type a pattern (e.g. ABC*) in the name area.

Subset Name _____

Direction	Options	Action
1 1. All	- IN-Clause...	1 1. Find
2. Next		2. Highlight
3. Previous		3. Scroll
4. First		4. Print
5. Last		5. Punch
		6. Exclude

Fields

Field	Description																														
Subset Name	<p>Specifies a predefined COBOL language subset name. Subset names may be concatenated by placing a plus sign (+) between the subset names.</p> <p>To display the Selection List pop-up, leave the Subset Name field blank, or type a pattern with wildcard characters (? for one character; * for zero or more characters). See "Selection List Pop-ups" on page 134 for more information.</p> <p>These are the predefined COBOL language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CAll</td><td>DIREctive</td><td>LABel</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTION</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table>	ASsignment	DEFinition	IO	CAll	DIREctive	LABel	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTION	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS	
ASsignment	DEFinition	IO																													
CAll	DIREctive	LABel																													
CIcs	DIVision	MATH																													
COBOLII	DL/I	Output																													
COBOL/370	DML	PARagraph																													
COMment	ENtry	PERform																													
CONditional	EXIt	SECTION																													
DB2 SQL	GOto	SORTMerge																													
DDL	IDMS	STructure																													
DEBug	IMS																														
Direction	<p>Specifies a direction to restrict the search to a specific direction or occurrence. Direction is not available in Tree View.</p> <ul style="list-style-type: none">• Next - Searches forward from the cursor position to the next occurrence of the subset.• Previous - Searches backward from the cursor position to the previous occurrence of the subset.• First - Searches from the top of the source file to the first occurrence of the subset.• Last - Searches backward from the bottom of the source file to the last occurrence of the subset.• All - Searches for all occurrences of the subset and displays the number found in the short/long message field. This is the default.																														

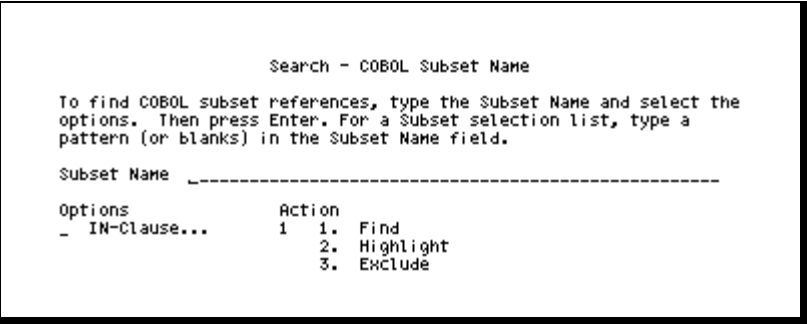
Field	Description
Options	Displays the Search - IN Clause pop-up used to restrict the source lines to be considered for a search. If left blank, the entire source code program is searched. The default is the entire program. See "Search - IN Clause Pop-up" on page 135 for more information.
Action	<p>Specifies the action to be performed on the subset. The default is Find. Scroll, Print, and Punch are not available in Tree View.</p> <ul style="list-style-type: none">• Find - Searches for one or all occurrences of the subset. You can specify DBCS identifiers or patterns to locate DBCS strings.• Highlight - Highlights source code lines containing the subset. Lines that are already highlighted are not reset.• Scroll - Positions the screen to the first line containing the subset. Highlighted lines remain unchanged. This action is available only on the editor screen. To scroll in Tree View, select View ► Scroll.• Print - Copies lines containing the subset to the List file. This action is available only on the editor screen. The List file is processed on the Options - Log/List/Punch Definition pop-up.• Punch - Copies lines containing the subset to the Punch file. This action is available only on the editor screen. The Punch file is processed on from the Options - Log/List/Punch Definition pop-up.• Exclude - Omits lines containing occurrences of the subset from the display.

Usage Notes

To find, highlight, or exclude statements corresponding to subset types while in Tree view, follow this step:

- ▶ Select View ▶ Scroll. The Search - COBOL Subset Name pop-in, shown in [Figure 92](#), displays.

Figure 92 • Search COBOL Subset Name Pop-up in Tree View



Note: _____

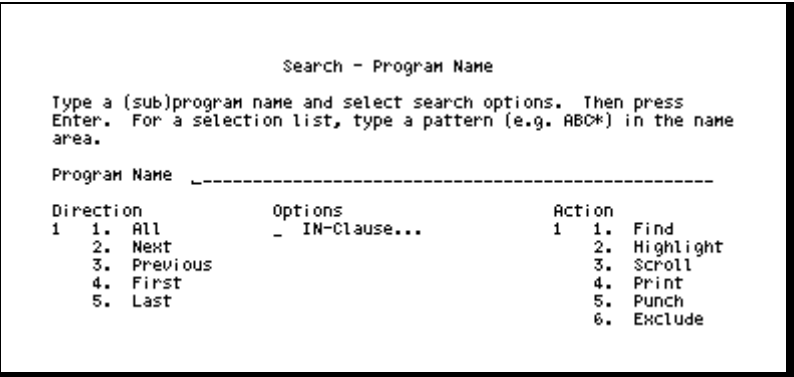
The Direction, Scroll, Print, and Punch actions are not available in Tree View.

Search - Program Name Pop-up

To find, highlight, scroll, print, punch, or exclude statements in subprograms, follow this step:

- ▶ Select Search ▶ Program. The Search - Program Name pop-up, shown in [Figure 93](#), displays.

Figure 93 • Search - Program Name Pop-up



Fields

Field	Description
Program Name	<p>Specifies the name of the main program or any nested program. Program names may be concatenated by placing a plus sign (+) between the program names.</p> <p>To display the Selection List pop-up, leave the Program Name field blank, or type a pattern with wildcard characters (? for one character; * for zero or more characters). See "Selection List Pop-ups" on page 134 for more information.</p>
Direction	<p>Specifies a direction to restrict the search to a specific direction or occurrence. Direction is not available in Tree View.</p> <ul style="list-style-type: none">• All - Searches for all occurrences of the program name and displays the number found in the short/long message field. This is the default.• Next - Searches forward from the cursor position to the next occurrence of the program name.• Previous - Searches backward from the cursor position to the previous occurrence of the program name.• First - Searches from the top of the source file to the first occurrence of the program name.• Last - Searches backward from the bottom of the source file to the last occurrence of the program name.

Field	Description
Options	Displays the Search - IN Clause pop-up used to restrict the source lines to be considered for a search. If left blank, the entire source code program is searched. The default is the entire program.
Action	<p>Specifies the action to be performed on the program name. The default is Find. Scroll, Print, and Punch are not available in Tree View. These are the valid actions:</p> <ul style="list-style-type: none"> • Find - Searches for one or all occurrences of the program name. You can specify DBCS identifiers or patterns to locate DBCS strings. This is the default. • Highlight - Highlights source code lines containing the program name. Lines that are already highlighted are not reset. • Scroll - Positions the screen to the first line containing the program name. Highlighted lines remain unchanged. This action is available only on the editor screen. To scroll in Tree View, select View ► Scroll. • Print - Copies lines containing the program name to the List file. This action is available only on the editor screen. The List file is processed on the Options - Log/List/Punch Definition pop-up. • Punch - Copies lines containing the program name to the Punch file. This action is available only on the editor screen. The Punch file is processed on the Options - Log/List/Punch Definition pop-up. • Exclude - Omits lines containing occurrences of the program name from the display.

Usage Notes

To find, highlight, or exclude statements in subprograms from Tree view, follow this step:

- ▶ Select View ▶ Scroll. The Search - Program Name pop-up, shown in [Figure 94](#), displays.

Figure 94 • Search - Program Name Pop-up in Tree View

Search - Program Name

To find all statements in a (sub)program, type the Program Name and select the options. Then press Enter. For a Name selection list, type a pattern (or blanks) in the Program Name field.

Program Name _____

Options	Action
- IN-Clause...	1 1. Find 2. Highlight 3. Exclude

Note: _____

The Direction, Scroll, Print, and Punch actions are not available in Tree View.

Search - Line Range Pop-up

To find, highlight, scroll, print, punch, or exclude statements in ranges of lines, follow this step:

- ▶ Select Search ▶ Line. The Search - Line Range pop-up, shown in [Figure 95](#), displays.

Figure 95 • Search - Line Range Pop-up

Search - Line Range

Type a line range (or line) and select search options. The press Enter.

Line Range _____

Direction	Options	Action
1 1. All 2. Next 3. Previous 4. First 5. Last	- IN-Clause...	1 1. Find 2. Highlight 3. Scroll 4. Print 5. Punch 6. Exclude

Fields

Field	Description
Line Range	Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor screen.
Direction	<p>Specifies a direction to restrict the search to a specific direction or occurrence. Direction is not available in Tree View.</p> <ul style="list-style-type: none">• All - Searches for all occurrences of the line range and displays the number found in the short/long message field. This is the default.• Next - Searches forward from the cursor position to the next occurrence of the line range.• Previous - Searches backward from the cursor position to the previous occurrence of the line range.• First - Searches from the top of the source file to the first occurrence of the line range.• Last - Searches backward from the bottom of the source file to the last occurrence of the line range.

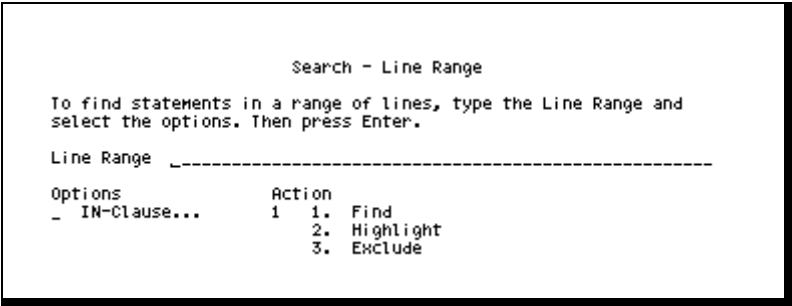
Field	Description
Options	Displays the Search - IN Clause pop-up used to restrict the source lines to be considered for a search. If left blank, the entire source code program is searched. The default is the entire program. See "Search - IN Clause Pop-up" on page 135 for more information.
Action	<p>Specifies the action to be performed on the line range. Scroll, Print, and Punch are not available in Tree View.</p> <ul style="list-style-type: none">• Find - Searches for one or all occurrences of the line range. You can specify DBCS identifiers or patterns to locate DBCS strings. This is the default.• Highlight - Highlights source code lines containing the line range. Lines that are already highlighted are not reset.• Scroll - Positions the screen to the first line containing the line range. Highlighted lines remain unchanged. This action is available only on the editor screen. To scroll in Tree View, select View ► Scroll.• Print - Copies lines containing the line range to the List file. This action is available only on the editor screen. The List file is processed on the Options - Log/List/Punch Definition pop-up.• Punch - Copies lines containing the line range to the Punch file. This action is available only on the editor screen. The Punch file is processed on the Options - Log/List/Punch Definition pop-up.• Exclude - Omits lines containing occurrences of the line range from the display.

Usage Notes

To find, highlight, or exclude lines from Tree View, follow this step:

- ▶ Select View ▶ Scroll. The Search - Line Range pop-up, shown in [Figure 96](#), displays.

Figure 96 • Search - Line Range Pop-up in Tree View



Note: _____

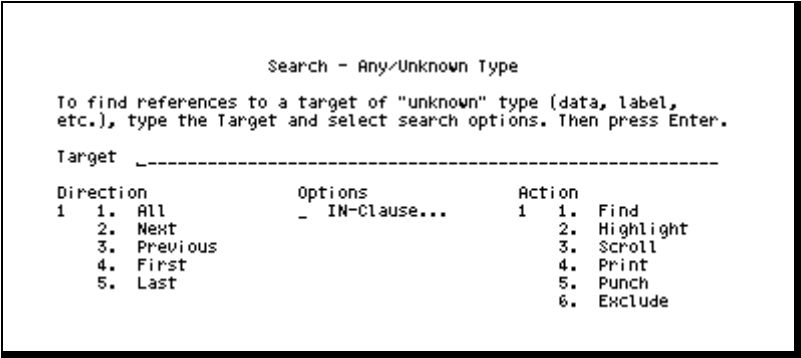
The Direction, Scroll, Print, and Punch actions are not available in Tree View.

Search - Any/Unknown Type Pop-up

To find, highlight, scroll, print, punch, or exclude targets when the type is unknown, follow this step:

- ▶ Select Search ▶ Any. The Search - Any/Unknown Type pop-up, shown in [Figure 97](#), displays.

Figure 97 • Search - Any/Unknown Type Pop-up



Fields

Field	Description
Target	Specifies the name of the target for that you want to search.
Direction	<p>Specifies a direction to restrict the search to a specific direction or occurrence. Direction is not available in Tree View.</p> <ul style="list-style-type: none">• Next - Searches forward from the cursor position to the next occurrence of the target.• Previous - Searches backward from the cursor position to the previous occurrence of the target.• First - Searches from the top of the source file to the first occurrence of the target.• Last - Searches backward from the bottom of the source file to the last occurrence of the target.• All - Searches for all occurrences of the target and displays the number found in the short/long message field. This is the default.

Field	Description
Options	Displays the Search - IN Clause pop-up used to restrict the source lines to be considered for a search. If left blank, the entire source code program is searched. The default is the entire program. See Figure 102 on page 135 for more information.
Action	<p>Specifies the action to be performed on the target. The default is Find. Scroll, Print, and Punch are not available in Tree View.</p> <ul style="list-style-type: none"> Find - Searches for one or all occurrences of the target. You can specify DBCS identifiers or patterns to locate DBCS strings. This is the default. Highlight - Highlights source code lines containing the target. Lines that are already highlighted are not reset. Scroll - Positions the screen to the first line containing the target. Highlighted lines remain unchanged. This action is available only on the editor screen. To scroll in Tree View, select View ► Scroll. Print - Copies lines containing the target to the List file. This action is available only on the editor screen. The List file is processed on the Options - Log/List/Punch Definition pop-up. Punch - Copies lines containing the target to the Punch file. This action is available only on the editor screen. The Punch file is processed on the Options - Log/List/Punch Definition pop-up. Exclude - Omits lines containing occurrences of the target from the display.

Usage Notes

To find, highlight, or exclude statements containing the specified target from Tree view, follow this step:

- ▶ Select View ▶ Scroll. The Search - Any/Unknown pop-up, shown in [Figure 98](#), displays.

Figure 98 • Search - Any/Unknown Type Pop-up from Tree View

```

Search - Any/Unknown Type

To find references to a target of "unknown" type (data, label,
etc.), type the Target and select the options. Then press Enter.

Target _____

Options      Action
- IN-Clause... 1 1. Find
                2. Highlight
                3. Exclude
  
```

Note: _____

The Direction, Scroll, Print, and Punch actions are not available in Tree View.

Search - Branch Request Pop-up

To position the cursor at the specified target, follow this step:

- ▶ Select Search ▶ Branch. The Search - Branch Request pop-up, shown in [Figure 99](#), displays.

Figure 99 • Search - Branch Request Pop-up

```

Search - Branch Request

Type the Target and select the Target Type. Then press
Enter. For a selection list, type a pattern in the
Target field.

Target PROC_____

Target Type
-- 1. Cursor Position
   2. Label
   3. Program
   4. Backup
  
```

Fields

Field	Description
Target	<p>Specifies a label name or program name. This field is ignored if Target Type 1 is selected.</p> <p>To display the Selection List pop-up, select Target Type and leave Target blank, or type a pattern with wildcard characters (? for one character; * for zero or more characters). See Figure 101 on page 134 for more information.</p>
Target Type	Specifies the number of the target type corresponding to the entry in the Target field.
Cursor Position	<p>Specifies that the target is determined by positioning the cursor on the desired target. When this Target type is selected, the editor screen is redisplayed to allow cursor placement.</p> <p>If the statement is a GO TO, PERFORM or internal CALL, the cursor is positioned to the target of that statement. If any other statement type is selected, the cursor is positioned to the next sequential statement to be executed. The PROCEDURE DIVISION label is located if the cursor is positioned outside the PROCEDURE DIVISION</p>
Label	<p>Positions the cursor to the specified paragraph or section name.</p> <p>If the specified name is not fully qualified and there are multiple paragraphs or sections with the specified name, the cursor is positioned on the first paragraph or section found. A message displays indicating the number of paragraphs or sections found.</p>
Program	Positions the cursor to the specified nested program.
Backup	Positions the cursor at the location from which the previous branch occurred.

Usage Notes

When accessed from Tree View, the Target Type 1 Cursor Position option is not available, as shown in [Figure 100](#).

Figure 100 • Search - Branch Request Pop-up in Tree View

```

Search - Branch Request

Type the Target and select the Target Type. Then press
Enter. For a selection list, type a pattern in the
Target field.

Target  _____

Target Type
--  1. Label
    2. Program
    3. Backup

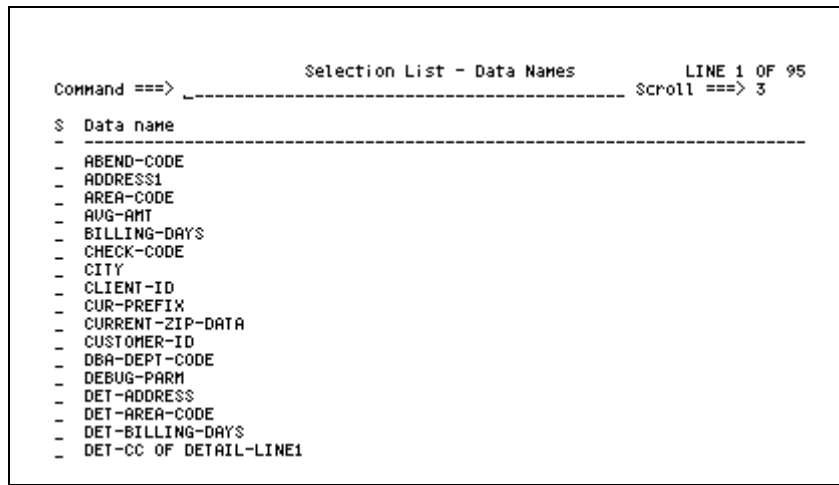
```

Selection List Pop-ups

Selection list pop-ups are available for the Data, Program, Label, Subset, and Branch searches. To display the selection list, leave the name field blank or type a pattern with wildcard characters (? for one character; * for zero or more characters) in the name field.

The selection list pop-ups are the same except for the type of list displayed. [Figure 101](#) is an example of the data names selection list.

Figure 101 • Search - Data Selection Pop-up



```

                                Selection List - Data Names          LINE 1 OF 95
Command ==> _____ Scroll ==> 3

$ Data name
-
- ABEND-CODE
- ADDRESS1
- AREA-CODE
- AVG-AMT
- BILLING-DAYS
- CHECK-CODE
- CITY
- CLIENT-ID
- CUR-PREFIX
- CURRENT-ZIP-DATA
- CUSTOMER-ID
- DBA-DEPT-CODE
- DEBUG-PARM
- DET-ADDRESS
- DET-AREA-CODE
- DET-BILLING-DAYS
- DET-CC OF DETAIL-LINE1

```

Use the S line command to select a name from the list. The Search pop-up redisplay with the selected name in the appropriate field.

You can select more than one name on these pop-ups:

- Selection List - Data Names
- Selection List - Label/Paragraph Names
- Selection List - Subset Names
- Selection List - Program Names

Search - IN Clause Pop-up

To restrict the source lines to be considered for a search, follow this step:

- ▶ Select Search ▶ IN-Clause. The Search - IN Clause pop-up, shown in [Figure 102](#), displays.

Figure 102 • Search - IN Clause Pop-up

Search - IN Clause

To restrict the source lines to be considered for a search, type one or more IN-Clause selections. Then press Enter. For a selection list, type a pattern in the appropriate input field (except Line Range).

Program Name . . .

Subset . . .

Label Name . . . P120-READ

Line Range . . .

Fields

Field	Description
Program Name	Specifies the name of the main program or any nested program. The search includes all code contained in the program, including all programs physically nested inside the specified program.
Subset	Specifies a predefined COBOL language subset name.
Note: See "Product Overview" on page 5 for a description of each subset.	
These are the predefined COBOL language subset names:	
ASsignment	DEFinition
CAll	DIREctive
CIcs	DIVision
COBOLII	DL/I
COBOL/370	DML
COMment	ENtry
CONditional	EXIt
DB2 SQL	GOto
DDL	IDMS
DEBug	IMS
	IO
	LABel
	MATH
	Output
	PARagraph
	PERform
	SECTion
	SORTMerge
	STructure

Field	Description
Label Name	Specifies a PROCEDURE DIVISION paragraph or section name. The PROCEDURE and PROC literals can also be specified.
Line Range	Specifies a single line number or range of lines.

Usage Notes

All fields on this pop-up are optional. If all fields are left blank, the search is not restricted; that is, the entire program is searched.

7

Check

This chapter describes the Check pull-down options and contains these sections:

Section	Page
Check Pull-down	137
Syntax Checking Process	139
Check - Compiler Options Pop-up	140
Check - Preprocessors Pop-up	143
Check - Modules and Libraries Pop-up	144

Check Pull-down

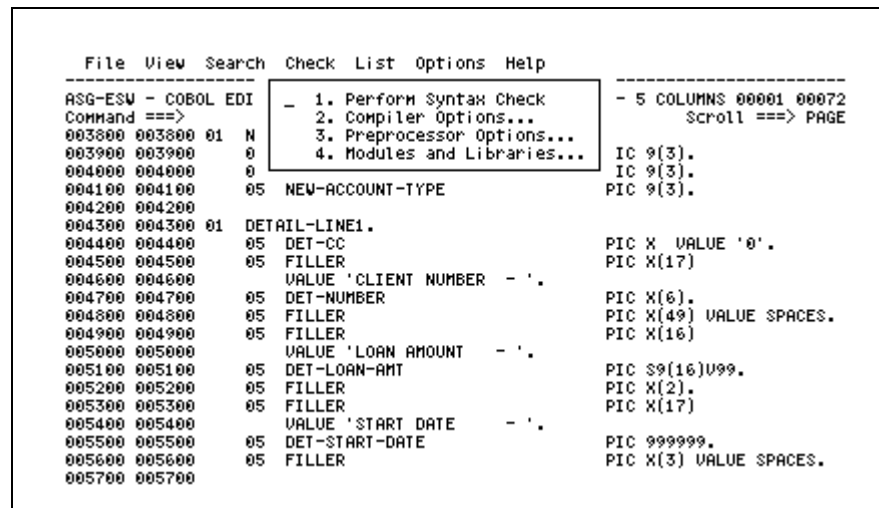
Note:

The Check action is not available in Tree View.

Selecting Check on the action bar displays the Check pull-down, shown in [Figure 103 on page 138](#). This pull-down provides an online syntax check mechanism for the COBOL source currently being edited.

SmartEdit performs a syntax check of the program and places syntax error messages directly below the line in which any errors are found.

Figure 103 • Check Pull-Down



Actions

Action	Description
Perform Syntax Check	Initiates the syntax checking process. See "Syntax Checking Process" on page 139 for more information.
Compiler Options	Displays the Check - Compiler Options pop-up to specify compiler options that affect the operation of the Check facility.
Preprocessor Options	Displays the Check - Preprocessors pop-up to select the preprocessors to be executed during the syntax check.
Modules and Libraries	Displays the Check - Modules and Libraries pop-up to specify the load modules and libraries to be used during the check process.

Syntax Checking Process

Selecting the Perform Syntax Check action on the Check pull-down begins the syntax checking process. Upon completion of this operation, a message displays stating the number of error(s) encountered.

To return the source file to its original unmarked form, follow this step:

- ▶ Select View ▶ Reset, or type RESET and press Enter.

The compiler dataset name is specified at the time that SmartEdit is installed and may be changed on the Check - Modules and Libraries pop-up. Copy libraries may be changed on the Options - COPY/Include Libraries screen by issuing the REFRESH command.

To view the errors, use the SCROLL HI NEXT and RSCROLL commands.

Check uses temporary files for its processing. If the check fails due to insufficient temporary file space, then increase the size of the Work File on the Options - Product Allocations pop-up.

To display the Options - Product Allocations pop-up, follow this step:

- ▶ Select Options ▶ Product Allocations, or type ALLOCDEF and press Enter.

For more information about the Options - Product Allocations pop-up, see ["Options - Product Allocations Pop-up" on page 164](#).

Check - Compiler Options Pop-up

To set options for the compiler used by the syntax check facility, follow this step:

- Select Check ► Compiler Options. The Check - Compiler Options pop-up, shown in [Figure 104](#), displays.

Figure 104 • Check - Compiler Options Pop-up

```

Check - Compiler Options

Options:          Flag Option:
- Migrate         2  1. Informational
- DBCS            2  2. Warning
/ Resident        3  3. Error
- Report Writer   4  4. Severe
- CMPR2           5  5. Unrecoverable
- MLE

DATEPROC Flag _  ('Y'es/'N'o)
YEARWINDOW ---- (Fixed : 1900 to 1999)
                  (Sliding: -1 to -99)

Apost/Quote:     Language:
1  1. Apost (')  1  1. English (mixed)
2  2. Quote (")  2  2. English (ucase)
                  3  3. Japanese (JP)
                  4  4. Japanese (JA)

Other Compiler Options:
-----

```

Fields

Field	Description
Options	<p>Specifies these compiler options:</p> <ul style="list-style-type: none"> • MIGRATE - Sets the MIGR or FLAGMIG option to flag statements requiring changes in order to function in the next level of the COBOL compiler. If you select Migrate, the Flag option should be set to display informational messages. • DBCS - Sets the DBCS flag for the COBOL compiler. • RESIDENT - Sets the compiler RES/NORES flag for the COBOL compiler. • Report Writer - Sets the COBOL II REPwtr flag for the COBOL compiler. • CMPR2 - Sets the COBOL II Release 3 compiler to function as a COBOL II Release 2 compiler. • MLE - Sets the Millennium Language Extension (MLE) flag for the COBOL compiler. If you select this compiler option, you must complete these options: <ul style="list-style-type: none"> — DATEPROC Flag - Enables and disables windowing with warnings ON or NOT. A Y in this field lets you receive the helpful diagnostic information. — YEARWINDOW - Sets the century window. You can choose the fixed window or sliding window method. If you do not specify a fixed or sliding value, the IBM default fixed value of 1950 is used. — Fixed - specify a 4-digit year between 1900 and 1999. — Sliding - select a negative integer from -1 to -99.

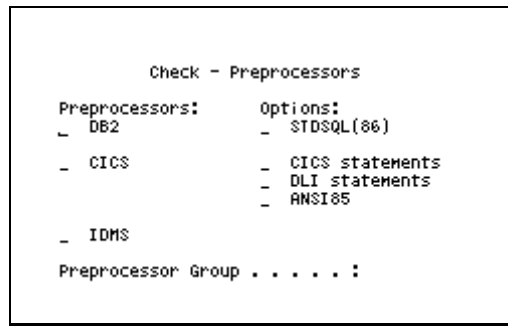
Field	Description
Flag Option	<p>Specifies the number of the minimum message level to be displayed by the syntax checking facility. The default is 2, display warning messages.</p> <ol style="list-style-type: none">1. Informational - Displays informational messages.2. Warning - Displays warning messages.3. Error - Displays error messages.4. Severe - Displays severe error messages.5. Unrecoverable - Displays unrecoverable error messages.
Apost/Quote	<p>Specifies whether the apostrophe (') or quotation (") symbol is used for syntax checking. The default is 1, apostrophe (').</p> <ol style="list-style-type: none">1. Apost (') - Uses the apostrophe for syntax checking. Select this option for COBOL II.2. Japanese (JP) - Sets the compiler language to Japanese. This option allows DBCS COBOL names and displays messages in English.3. Japanese (JA) - Sets the compiler language to Japanese. This option allows DCBC COBOL names and displays messages in KANJI.

Check - Preprocessors Pop-up

To select the preprocessors and preprocessor parameters to be used by the syntax check facility, follow this step:

- Select Check ► Preprocessor Options or select Preprocessor on the Check - Parameters pop-up. The Check - Preprocessors pop-up, shown in [Figure 105](#), displays.

Figure 105 • Check - Preprocessors Pop-up



Fields

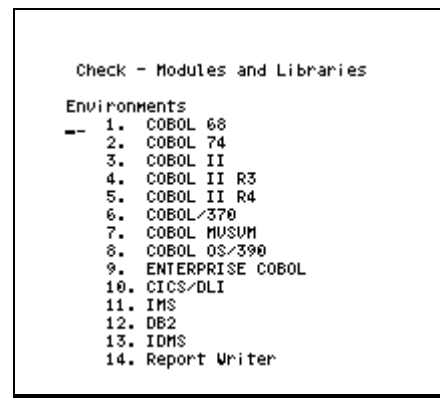
Operand	Description
Preprocessors	<p>Specifies a non-blank character to select the appropriate preprocessor.</p> <ul style="list-style-type: none"> • DB2 - If the source contains DB2 statements, select DB2. • CICS - If the source contains CICS statements, select CICS. • IDMS - If the source contains CA-IDMS statements, select IDMS.
Options	<p>Specifies a non-blank character to select the appropriate options for the corresponding preprocessor.</p> <ul style="list-style-type: none"> • STDSQL - If the DB2 preprocessor is selected, choose STDSQL(86) to process SQL statements as ANSI SQL 86. • CICS statements - If the CICS preprocessor is selected and the source contains CICS statements, select CICS. • DLI statements - If the CICS preprocessor is selected and the source contains DLI statements, select DLI. • ANSI85 - If the CICS preprocessor is selected and the source contains ANSI85 statements, select ANSI85. • Preprocessor Group: - Displays the name of the user-defined preprocessor group specified on the Options - Preprocessors pop-up, if a group has been specified.

Check - Modules and Libraries Pop-up

To select the environment for which load modules and libraries can be specified, follow this step:

- 1 Select Check ► Modules and Libraries and press Enter. The Check - Modules and Libraries pop-up, shown in [Figure 106](#), displays.

Figure 106 • Check - Modules and Libraries Pop-up



- 2 Type the number of the appropriate environment and press Enter.

Environments

Specifies the load modules and libraries for the check process.

Environment	Description
COBOL 68	Displays the Check - COBOL 68 pop-up to specify the COBOL 68 load module and load library.
COBOL 74	Displays the Check - COBOL 74 pop-up to specify the COBOL 74 load module and load library.
COBOL II	Displays the Check - COBOL II pop-up to specify the COBOL II load module and load library.
COBOL II R3	Displays the Check - COBOL II R3 pop-up to specify the COBOL II load module and load library.
COBOL II R4	Displays the Check - COBOL II R4 pop-up to specify the COBOL II load module and load library.
COBOL/370	Displays the Check - COBOL/370 pop-up to specify the COBOL/370 load module and load library.

Environment	Description
COBOL MVSVM	Displays the Check - COBOL MVSVM pop-up to specify the COBOL MVSVM load module and load library.
COBOL OS/390	Displays the Check - OS/390 pop-up to specify the OS/390 COBOL load module and load library.
Enterprise COBOL	Displays the Check - Enterprise COBOL pop-up to specify the Enterprise COBOL load module and library.
CICS/DLI	Displays the Check - CICS and DLI pop-up to specify the CICS/DLI load module and load library.
IMS	Displays the Check - IMS pop-up to specify the IMS load library.
DB2	Displays the DB2 Parameter Definition pop-up to specify the DB2 Load Module and Load Library, Subsys Name, Plan Name, and Authorization ID.
IDMS	Displays the IDMS Parameter Definition pop-up to specify CA-IDMS parameters, load libraries, and local mode dictionaries.
Report Writer	Displays the Check - Report Writer pop-up to specify the COBOL II Report Writer load library.

8

List

This chapter describes the List pull-down options and contains these sections:

Section	Page
List Pull-down	148
List - CALL Statements Pop-up	149
List - Equates Pop-up	150
List - Perform Range Names Pop-up	151
List - Program/Subprogram Names Pop-up	152
List - COBOL Subset Names Pop-up	153

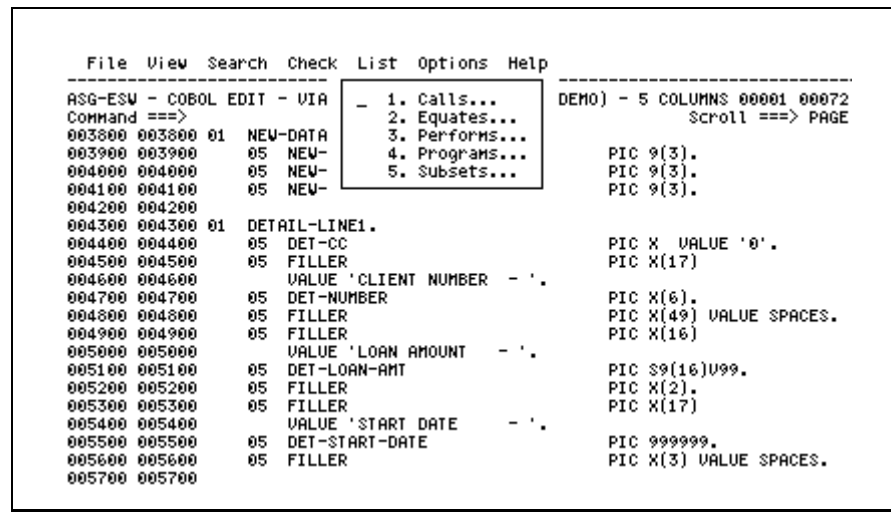
List Pull-down

Note:

The List action is not available in Tree View.

Selecting List on the action bar displays the List pull-down, shown in [Figure 107](#). Use this pull-down to access pop-ups that list information about CALLs, equates, PERFORMs, programs, and subsets.

Figure 107 • List Pull Down



Actions

Action	Description
Calls	Displays the List - CALL Statements pop-up that lists programs that are CALLED by the active program.
Equates	Displays the List - Equates pop-up that lists all equates for the active program.
Performs	Displays the List - Perform Range Names pop-up that lists all COBOL PERFORM ranges in the active program.
Programs	Displays the List - Program/Subprogram Names pop-up that lists the internal subprograms defined within a COBOL II Release 3 or later program.
Subsets	Displays the List - COBOL Subsets names pop-up that lists the COBOL subsets along with a brief description of each.

List - CALL Statements Pop-up

To display the programs that are CALLED by the active program, follow this step:

- Select List ► Calls or type LIST CALLS on the editor screen and press Enter. The List - CALL Statements pop-up, shown in [Figure 108](#), displays.

You can enter the LPRINT * command on this pop-up to copy the list of called programs to the List file.

Figure 108 • List - CALL Statements Pop-up

```

                                List - CALL Statements
Command ==> _____ Scroll ==> 3

$ CALL Statements
-----
- 'ABENDPGM'
- 'DBACLSSE1'
- 'DBACLSSE2'
- 'DBAOPEN1'
- 'DBAOPEN2'
- 'DBAREAD1'
- 'DBAREAD2'
- 'VIAFDEM1'
***** BOTTOM OF DATA *****

```

Fields

Field	Description
Line command area	Refers to the area to the left of the CALL Statements field, which accepts the S line command to select a specified CALL statement. Place the cursor on the called module name in the source statement.
CALL Statements	Lists the names of all programs that are called by the current program.

List - Equates Pop-up

To display all equates for the active program, follow this step:

- ▶ Select List ▶ Equates or type LIST EQUATES on the editor screen and press Enter. The List - Equates pop-up, shown in [Figure 109](#), displays.

You can enter the LPRINT * command on this pop-up to copy the list of equates to the List file.

Figure 109 • List - Equates Pop-up

```

-----
                                List - Equates
Command ==>_____Scroll ==> CSH

    Name          Substitution String
-----
COUNTRY          USA
ZIPCODE          AAA
*****
*****BOTTOM OF DATA*****

```

Fields

Field	Description
Name	Specifies the name assigned to the equate.
Substitution String	Specifies the actual data the Name represents. Prior to execution, this string is substituted into any command that uses the Equate name. This field is left and right scrollable to show the entire string. The right-most column contains a greater than (>) symbol when the substitution string extends past the screen column capability.

List - Perform Range Names Pop-up

To display all COBOL perform ranges in the program being edited, follow this step:

- Select List ► Performs or type LIST PERFORMS on the editor screen and press Enter. The List - Perform Range Names pop-up, shown in [Figure 110](#), displays.

You can enter the LPRINT * command on this pop-up to copy the list of perform range names to the List file.

Figure 110 • List - Perform Range Names Pop-up

```

                                List - Perform Range Names
Command ==> _____ Scroll ==> 3

$ Perform Range Name
-----
- PROGRAM-INIT
- P000-NEXT THRU P000-EXIT
- P005-UAL-PARM THRU P005-EXIT
- P010-OPEN THRU P010-EXIT
- P100-PRINT THRU P119-EXIT
- P120-READ THRU P129-EXIT
- P150-SUBTOT THRU P169-EXIT
- P155-CL-SUBTOT THRU P159-EXIT
- P160-HDG THRU P169-EXIT
- P999-ABEND-PROGRAM OF ABEND-PROGRAM
***** BOTTOM OF DATA *****

```

Fields

Field	Description
Line command area	Refers to the area to the left of the Perform Range Name field, which accepts the S line command to select any perform range listed. Place the cursor on the label for the first paragraph in the perform range.
Perform Range Name	Specifies the perform range name.

List - Program/Subprogram Names Pop-up

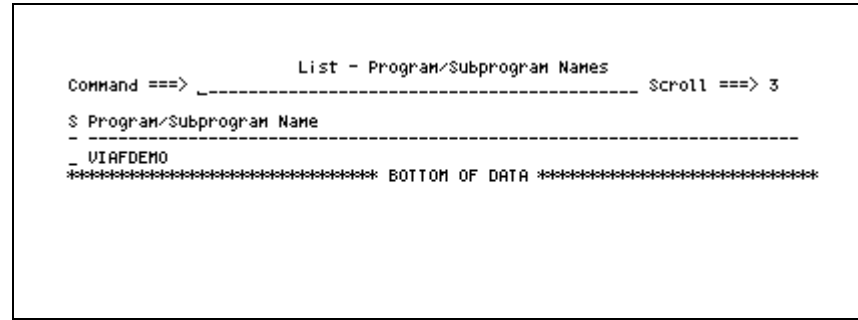
The List - Program/Subprogram Names pop-up displays the internal subprograms defined within a COBOL II Release 3 or later program. The internal subprograms are indented in relation to their hierarchy.

To display this pop-up, follow this step:

- Select List ► Programs or type LIST PROGRAMS on the editor screen and press Enter. The List - Program.Subprogram Names pop-up, shown in [Figure 111](#), displays.

You can enter the LPRINT * command on this pop-up to copy the list of programs to the List file.

Figure 111 • List - Program/Subprogram Names Pop-up



Fields

Field	Description
Line command area	Refers to the area to the left of the Program/Subprogram Name field, which accepts the S line command to select a program. Place the cursor on the first statement in the specified program.
Program/Subprogram Name	Specifies the program label name. For COBOL II Release 3 or later programs, this includes the main program and all subprograms.

List - COBOL Subset Names Pop-up

To displays the subsets along with a brief description of each, follow this step:

- Select List ► Subsets or type LIST SUBSETS on the editor screen and press Enter.
The List - COBOL Subset Name pop-up, shown in [Figure 112](#), displays.

The subsets listed on this pop-up include COBOL language subsets, screen subsets, and tagged lines subsets. Various SmartEdit commands accept subsets as operands, which limit their scope to a particular type of data.

You can enter The LPRINT * command on this pop-up to copy the list of subsets to the List file.

Figure 112 • List - COBOL Subset Names Pop-up

List - COBOL Subset Names	
Command ==> _____	Scroll ==> 3
Subset	Description
- AS(signment)	- COBOL: ACCEPT, ADD, SUBTRACT, MULTIPLY, DIVIDE, BY, COMPUTE, MOVE, FROM, INSPECT, EXAMINE, STRING, UNSTRING, SET, SEARCH, TRANSFORM. IDMS: ACCEPT, CHECK, ERASE, GET, INQUIRE, LOAD, PUT, RETURN. SQL: FETCH, SELECT INTO, SET.
- CA(11)	- COBOL: CALL, CANCEL, ENTRY, intrinsic function calls. CICS: LINK, XCTL. IDMS: ATTACH, LINK, XCTL.
- CI(cs)	- EXEC CICS and EXEC DL/I commands.
- COBOLII	- CALL (COBOLII), PERFORM (COBOLII), CONTINUE, SET (to TRUE), SERVICE LABEL, EVALUATE, INITIALIZE, END VERBS.
- COBOL370	- Statements and clauses unique to COBOL/370 such as intrinsic function calls, procedure pointers and calls to the LE/370 run-time environment.
- COM(ment)	- Comment lines and Identification division.

Fields

Field	Description
Line command area	Refers to the area to the left of the Subset column, which accepts the S line command to select the subset. Place the cursor on the first statement in the specified subset.
Subset	Specifies the subset name.
Description	Specifies the description of the subset.

See ["Product Overview" on page 5](#) for more information and a list of the COBOL subsets and their descriptions.

9

Options

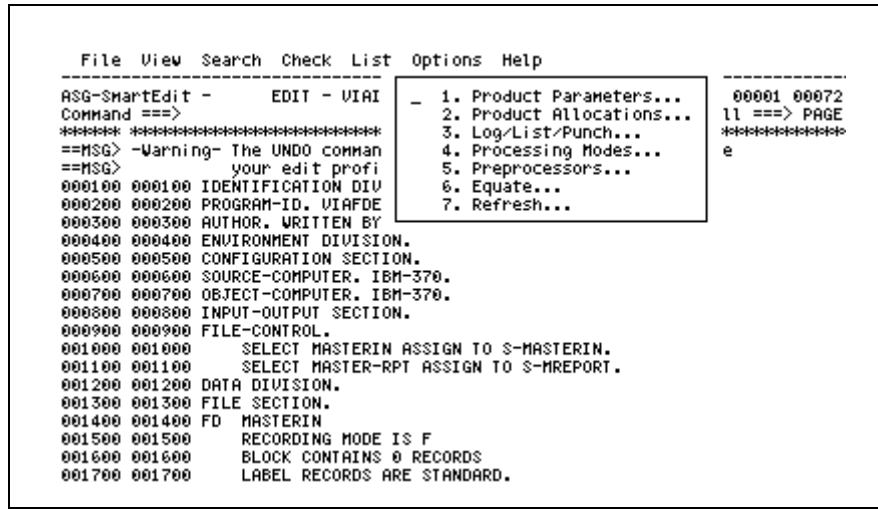
This chapter describes the Options pull-down and contains these sections:

Section	Page
Options Pull-down	156
Options - Product Parameters Pop-up	157
Options - Product Allocations Pop-up	164
Options - Log/List/Punch Definition Pop-up	165
Options - Processing Modes Pop-up	169
Options - Preprocessors Pop-up	172
Options - Equate Pop-up	174

Options Pull-down

Selecting Options on the action bar displays the Options pull-down (see [Figure 113](#)). Use this pull-down to customize your SmartEdit environment by setting certain parameters and options, setting processing modes, defining equates, or refreshing the display.

Figure 113 • Options Pull-Down



Actions

Action	Description
Product Parameters	Displays the Options - Product Parameters pop-up used to set parameters that affect the online operation of SmartEdit.
Product Allocations	Displays the Options - Product Allocations pop-up used to specify DASD volumes for the Log, List, Punch, and Work files.
Log/List/Punch	Displays the Options - Log/List/Punch Definition pop-up used to set values for allocating, formatting, and processing the SmartEdit Log, List, and Punch files.
Processing Modes	Displays the Options - Processing Modes pop-up used to set processing modes such as LEARN, XMODE, IDMS, and DB2.
Preprocessors	Displays the Options - Preprocessors pop-up used to select a preprocessor group name.

Action	Description
Equate	Displays the Options - Equate pop-up used to define a name for a character string.
Refresh	<p>Redisplays the Options - COPY/Include Libraries screen so you can update the Copy Library information, then reads the most recent COPY members. COPY members are retrieved from partitioned datasets, and Librarian and Panvalet datasets. The ++INCLUDE lines are retrieved from Panvalet datasets only. The -INC lines are retrieved from Librarian datasets only.</p> <p>When editing a Panvalet or Librarian member, the ++ INCLUDE or -INC expansion is set to NO. This allows SmartEdit to do the expansion when necessary.</p>

See ["Getting Started" on page 21](#) for descriptions of fields on the Options - COPY/Include Libraries screen ([Figure 11 on page 24](#)).

Options - Product Parameters Pop-up

To set parameters for database support that affects the online operation of SmartEdit, follow this step:

- Select Options ► Product Parameters or type PARMDEF on any screen and press Enter. The Options - Product Parameters pop-up, shown in [Figure 114](#), displays.

Figure 114 • Options - Product Parameters Pop-up

```

Options - Product Parameters

ALARM . . . . . YES (Yes or No)
Cursor Character % (Token sub. character)

Data Base Parameters
-- 1. DB2
   2. IDMS
   3. DATACOM

```

Fields

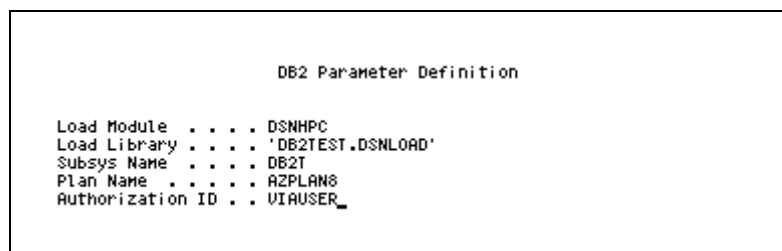
Field	Description
Alarm	Controls the audible alarm on the terminal. If YES, the alarm sounds when an error message displays. The default is YES.
Cursor Character	Sets the cursor token substitution character. The default is percent (%). Cursor substitution saves time by allowing you to use this character in a command and then place the cursor under the token in the source code. See "Commands" on page 181 for more information.
Database Parameters	Specifies the databases required for your SmartEdit session. <ol style="list-style-type: none"> 1. DB2 Support - Displays the DB2 Parameter Definition pop-up to specify the DB2 parameters. 2. IDMS - Displays the IDMS Parameter Definition pop-up to specify the CA-IDMS parameters. 3. DATACOM - Displays the DATACOM Parameter Definition pop-up to specify the DATACOM parameters.

DB2 Parameter Definition Pop-up

To set parameters for DB2 support, follow this step:

- Select the DB2 option on the Options - Product Parameters pop-up and press Enter. The DB2 Parameter Definition pop-up, shown in [Figure 115](#), displays.

Figure 115 • DB2 Parameter Definition Pop-up



```

                                DB2 Parameter Definition

Load Module . . . . DSNHPC
Load Library . . . . 'DB2TEST.DSNLOAD'
Subsys Name . . . . DB2T
Plan Name . . . . AZPLAN6
Authorization ID . . UIAUSER_

```

Note:

The DB2 parameters default to those specified during product installation.

Fields

Enter or verify this information for DB2 support, if necessary.

Operand	Description
Load Module	Specifies the name of the DB2 load module.
Load Library	Specifies the name of the DB2 load library.
Subsystem Name	Specifies the name of the DB2 subsystem.
Plan Name	Specifies the name of the DB2 plan.
Authorization ID	Specifies the authorization ID needed to access DB2 libraries. The default is your user ID.

IDMS Parameter Definition Pop-up

To set parameters for CA-IDMS support, follow this step:

- Select the IDMS option on the Options - Product Parameters pop-up and press Enter. The IDMS Parameter Definition pop-up, shown in [Figure 116](#), displays.

Figure 116 • IDMS Parameter Definition Pop-up

```

IDMS Parameter Definition

DB Name . . . . . UATDICT
Node . . . . . -----
SYSCTL Library . . . 'IDMS.PROD10.SYSCTL'

DMCL Name . . . . . R120DMCL
Use SYSIDMS Load Module defaults NO      (Yes or No)

Libraries
-- 1. Load Libraries
   2. Local Mode Dictionaries

```

Note:

The CA-IDMS parameters default to those specified during product installation.

Fields

Enter or verify this information for CA-IDMS support, if necessary:

Field	Description
DB	Specifies the Data Base Dictionary name.
Node	Specifies the CV that controls the Dictionary to be accessed.
SYSCTL Library	Specifies the name of the SYSCTL library.
DMCL Name	Specifies the name of the IDMS DMCL module to be used when accessing the IDMS central version. The default value is R120DMCL. Enter a specific user value or blanks to request the IDMS default value. This field is only used for IDMS Release 12.0 and higher.
Use SYSIDMS Load Module defaults	<p>Indicates whether the default values from the IDMS SYSIDMS load module should be used. The IDMS SYSIDMS load module must reside in one of the IDMS load libraries specified using the load libraries selection. Specify Yes or No.</p> <p>If Yes is specified in this field, the DMCL name is ignored. This field is only used for IDMS Release 12.01 and higher.</p>
Libraries	<p>Specifies the type of IDMS library to be defined.</p> <ol style="list-style-type: none"> 1. Load Libraries - Displays the Options - IDMS Load Module Libraries pop-up to specify the names of load libraries to be used for IDMS programs. 2. Local Mode Dictionaries - Displays the Options - IDMS Local Mode Dictionaries pop-up to specify the IDMS local mode dictionaries to be used for processing IDMS programs.

IDMS Load Module Libraries Pop-up

To specify the names of load libraries to be used for IDMS programs, follow this step:

- ▶ Select the Load libraries option on the IDMS Parameter Definition pop-up and press Enter. The Options - IDMS Load Module Libraries, shown in [Figure 117](#), displays.

Figure 117 • Options - IDMS Load Module Libraries

```
Options - IDMS Load Module Libraries
Command ==> _____ Scroll ==> 3
Specify IDMS Load Libraries.
      Data Set Name
- -----
- 'IDMS.PROD10.LOADLIB'
- 'VIA60.PROD.XALOAD'
- 'VIA60.DEVL.XALOAD'
- 'VIA60.PROD.XALOAD'
- 'VIA60.PROD.XALOAD'
- 'VIA60.PROD.XALOAD'
- 'VIA60.PROD.XALOAD'
***** BOTTOM OF DATA *****
```

Note:

The IDMS Load Library dataset names specified at installation are listed by default.

Fields

Field	Description
Line command area	Refers to the area to the left of the Data Set Name field, which is used to insert and delete IDMS load libraries. The line command area accepts these line commands: <ul style="list-style-type: none"> • I Inserts a new line to add a new load library. • D Deletes a load library. • R Repeat a line (for modification).
Data Set Name	Specifies the name of the IDMS load library. Type the dataset name(s) to be used while processing IDMS programs. The list is scrollable.

IDMS Local Mode Dictionaries Pop-up

To specify the IDMS local mode dictionaries to be used for processing IDMS programs, follow this step:

- ▶ Select the Local Mode Dictionaries option on the IDMS Parameter Definition pop-up and press Enter. The Options - IDMS Local Mode Dictionaries pop-up, shown in [Figure 118](#), displays.

Figure 118 • Options - IDMS Local Mode Dictionaries Pop-up

```
Options - IDMS Local Mode Dictionaries
Command ==> _----- $scroll ==> 3
Specify IDMS Local Mode Dictionaries and DD Names.
  DD Name  Data Set Name
  -----
- DICTDB  'IDMS.PROD10.DICTDB'
- DLODDB  'IDMS.PROD10.DLODDB'
***** BOTTOM OF DATA *****
```

Note:

The IDMS Local Mode Dictionaries specified at installation are listed by default.

Fields

Field	Description
Line command area	Refers to the area to the left of the DD Name field, which is used to insert and delete IDMS local mode dictionaries. The line command area accepts these line commands: <ul style="list-style-type: none"> • I Inserts a new line to add a new local mode dictionary. • D Deletes a local mode dictionary. • R Repeat a line (for modification).
DD Name	Specifies the DDname to be used for the dataset.
Data Set Name	Specifies the dataset name(s) to be used while processing IDMS programs. The list is scrollable.

DATAKOM Parameter Definition Pop-up

To specify the parameters for DATAKOM support, follow this step:

- ▶ Select the DATAKOM option on the Options - Product Parameters pop-up and press Enter. The DATAKOM Parameter Definition pop-up, shown in [Figure 119](#), displays.

Figure 119 • DATAKOM Parameter Definition Pop-up

```

DATAKOM Parameter Definition
User Access Name . . . 
Data Base ID . . . 
User Access Pswd . . .
  
```

Fields

Type this information for DATAKOM support, if necessary:

Field	Description
User Access Name	Specifies the user access name.
Data Base ID	Specifies the database ID.
User Access Password	Specifies the user access password.

Options - Product Allocations Pop-up

To specify the DASD volumes for the Log, List, Punch, and Work files, and to specify disk storage space for the Work file, follow this step:

- Select Options ► Product Allocations or type ALLOCDEF on any screen. The Options - Product Allocations pop-up, shown in [Figure 120](#), displays.

Figure 120 • Options - Product Allocations Pop-up

```

Command ==> Options - Product Allocations
-----
Log file:
Generic unit . . . SRTDA      (generic group name or unit address)
Volume serial . . SRT801    (blank for authorized default volume)
List file:
Generic unit . . . SYSDA      (generic group name or unit address)
Volume serial . . SRT801    (blank for authorized default volume)
Punch file:
Generic unit . . . SRTDA      (generic group name or unit address)
Volume serial . . SRT801    (blank for authorized default volume)
Work file:
Generic unit . . . SYSDA      (generic group name or unit address)
Volume serial . . -----    (blank for authorized default volume)
Space units . . . CYLS       (BLKS, TRKS or CYLS)
Primary space . . 5          (space units)
Secondary space  5          (space units)

```

Fields

Field	Description
Log file	Specifies either the Management Class and Storage Class or Generic unit and volume serial number for the Log file that is allocated upon entry into SmartEdit. The Log file is used for error messages and log commands. File characteristics are specified on the Options - Log/List/Punch Definition pop-up.
List file	Specifies the Management Class and Storage Class or the Generic unit and volume serial number for the List file that is allocated the first time a request is made to print output. The List file is used for all printed output. File characteristics are specified on the Options - Log/List/Punch Definition pop-up.

Field	Description
Punch file	Specifies the Management Class and Storage Class or Generic unit and Volume serial number for the Punch file that is allocated the first time a request is made to punch output. The Punch file is used for all punched output. File characteristics are specified on the Options - Log/List/Punch Definition pop-up.
Work file	Specifies the Management Class, Storage Class, and Data Class or the unit and space requirements for the Work file that is allocated on entry into Conversion Assistant. The Work file is a temporary file. Consider the size of the programs you are working with when determining space requirements. Approximately one cylinder (on a 3380 device) is required for a 5,000 line program.

Options - Log/List/Punch Definition Pop-up

To set values for allocating, formatting, and processing the SmartEdit Log, List, and Punch files, follow this step:

- Select Options ► Log/List/Punch or type PRINTLOG or PRINTLST on any screen. The Options - Log/List/Punch Definition pop-up, shown in [Figure 121](#), displays.

Figure 121 • Options - Log/List/Punch Definition Pop-up

```

Options - Log/List/Punch Definition
Command ==> _____
1 - Process log file  2 - Process list file  3 - Process punch file
4 - Customized data set name

Options              Log              List              Punch
-----
Process option      . . . . . K              PK              PK
Primary tracks      . . . . . 1              1              1
Secondary tracks    . . . . . 2              5              5
Lines per page      . . . . . 56             56             56
Sysout class        . . . . . *              *              *

Process options: PK (print/keep), PD (print/delete), K, or D.

Job statement information:
//USER1 JOB (DEV123,263200,SRT,00),'USER1',PTY=6,
//      MSGCLASS=X
//*JOBPARM SYSAFF=CPUC
//*

```

Volumes for the Log, List, and Punch files are specified on the Options - Product Allocations pop-up.

Options

Option	Description
1 - Process log file	<p>Processes the options for the Log file. A new file is allocated to collect additional data, if required. The file name is in the format:</p> <p style="text-align: center;"><code>userid.SEDxxxxx.VIALOG</code></p> <p>where <code>xxxxx</code> is a unique number assigned by SmartEdit.</p> <p>If you specify the PK or PD Process option, Job statement information must be entered prior to selecting option 1. The Log file is processed when you press Enter. It is not necessary to exit SmartEdit to process the Log file.</p>
2 - Process List file	<p>Processes the options for the List file. A new file is allocated to collect additional data, if required. The file name is in the format:</p> <p style="text-align: center;"><code>userid.SEDxxxxx.VIALIST</code></p> <p>where <code>xxxxx</code> is a unique number assigned by SmartEdit.</p> <p>If the PK or PD Process option is specified, Job statement information must be entered prior to selecting option 2. The List file is processed when you press Enter. It is not necessary to exit SmartEdit to process the List file.</p>
3 - Process punch file	<p>Processes the options for the Punch file. This file is defined as fixed block with 80 byte records. The file name is in the format:</p> <p style="text-align: center;"><code>userid.SEDxxxxx.VIAPUNCH</code></p> <p>where <code>xxxxx</code> is a unique number assigned by SmartEdit.</p> <p>If the PK or PD Process option is specified, Job statement information must be entered prior to selecting option 3. The Punch file is processed when you press Enter. It is not necessary to exit SmartEdit to process the Punch file.</p>
4 - Customized data set name	<p>Displays the Options - Log/List/Punch Name Customization pop-up (Figure 122 on page 167), which enables you to specify a dataset where these files will be saved.</p>

Customizing Log/List/Punch Dataset Names

If you specified the K or PK process option on the Options - Log/List/Punch Name Customization pop-up, shown in [Figure 122](#), you can customize the dataset where the log, list, or punch file is allocated. By default, these files are allocated as:

`userid.yyyxxxxx.VIAnnnnn`

where:

`yyy` is the product ID

`xxxx` is a sequential number from 00001 to 99999

`nnnn` is LOG for Log, LIST for List, and PUNCH for Punch files

If you specified a TSO prefix, the prefix is appended to the beginning of the file name allocated.

Figure 122 • Options - Log/List/Punch Name Customization Pop-up

```

Command ==> Options - Log/List/Punch Name Customization
-----
You can define a customized name when you choose option PK (print/keep) or
K (keep) by specifying U(ser). Specifying Y(es) on Prompt later lets you
define a custom name as you process each log/list/punch file. Otherwise,
specify data set name and file mode. Then press Enter.

Options          Log      List      Punch
-----
File Naming . . . . . U      U      S      U(ser) or S(ystem)
Prompt later for DSN . . Y      Y      -      Y(es) or N(o)

The following are needed if U(ser) and N(o) are specified above

Log Data set name ----- (Seq)
File Mode . . 0      0(overwrite) or A(ppend)
List Data set name ----- (Seq)
File Mode . . -      0(overwrite) or A(ppend)
Punch Data set name ----- (Seq)
File Mode . . -      0(overwrite) or A(ppend)

```

Type U in the File Naming field for Log, List, and/or Punch to indicate user-defined a dataset name. If you specify N in the Prompt later for DSN field, you must type a dataset name in the corresponding dataset name field, and specify Overwrite or Append in the File Mode field.

Fields

Field	Description
Process option	Specifies one of the valid processing options. The default is PD for the Log and List files, and PK for the Punch file.
Primary tracks	Specifies the number of primary tracks to allocate. A size change does not take effect until the next allocation occurs. The default is 1.
Secondary tracks	Specifies the number of secondary tracks to allocate. A size change does not take effect until the next allocation occurs. Defaults are: 2 for the Log file, 5 for the List file, and 5 for the Punch file.
Lines per page	Specifies the number of lines to print per page. Standard maximum values are 60 for 6 lines per inch and 80 for 8 lines per inch. The default is 56.
Sysout class	Specifies the SYSOUT class value. The default is *, which sends the SYSOUT to the destination identified in the MSGCLASS parameter on the JOB statement.
Process options	Lists the options available for the Log, List, and Punch files. You may choose: <div> <div>PK - Print and keep</div> <div>K - Keep without printing</div> <div>PD - Print and delete</div> <div>D - Delete without printing</div> </div>
xxxxxx FILE IS ALLOCATED	Displays when the Log, List, or Punch file has been properly allocated. If the message does not display, check the assignments on the Options - Product Allocations pop-up. xxxxxx is LOG, LIST, or PUNCH depending on the file that was allocated.
Job statement information	Requires the correct job statement information for your installation. These JCL statements are required if the PK or PD process option has been specified.

Options - Processing Modes Pop-up

To set processing modes for SmartEdit, follow this step:

- Select Options ► Processing Modes. The Options - Processing Modes pop-up, shown in [Figure 123](#), displays.

Figure 123 • Options - Processing Modes Pop-up

Options - Processing Modes		
Command ==>	_____	Scroll ==> 3
Type option settings. Then press PF3/15 (END) to save options and exit.		
Option	Set	Description
CHANGEMAN	OFF	Allow ASG-SmartEdit to import Copy Libraries from Change M
CUA	ON	Enable CUA Action Bar
DB2	OFF	Allow ASG-SmartEdit to access the DB2 System Catalog
IDMS	OFF	Allow ASG-SmartEdit to access the IDMS Data Dictionary
LEARN	OFF	Display internally generated Primary Commands
LOCAL	OFF	Allow ASG-SmartEdit to access IDMS in Local Mode
XMODE	ON	Exclude all lines before executing Primary Commands

You can change a setting by typing over it, or by using the SET command. See ["SET Command" on page 272](#) for more information about the SET command.

Fields

Field	Description
Option	Specifies the processing mode to be enabled or disabled. The options include:
CHANGEMAN	<p>Allows SmartEdit to extract the copy libraries associated with a selected ChangeMan package and automatically populate the SmartEdit copy library screen. The extracted libraries are labeled {CHG}.</p> <ul style="list-style-type: none"> Libraries extracted from ChangeMan cannot be modified, moved, or deleted. Libraries extracted from ChangeMan can be copied or repeated, but the ChangeMan attribute is not passed on to replicated libraries (i.e., they do not have the {CHG} label and you can modify, move, or delete them). You may add your copy libraries to the list using the LOAD COPY LIST or EXTRACT COPY LIST commands. Regardless of the insertion position, all user libraries are processed before ChangeMan libraries. During a REFRESH, the libraries are listed in the order processed. <p>ChangeMan support is only available when these criteria are met:</p> <ul style="list-style-type: none"> You enter the SmartEdit session through the execution of the ChangeMan product. SmartEdit is available when ISPF Edit is invoked. ChangeMan EXIT 36 is modified to invoke SmartEdit as the primary editor under ChangeMan. SmartEdit uses TSO profile variables to determine if the ChangeMan environment is available. The STGLIB and PKGLVL variables must be present in the shared profile pool prior to invoking SmartEdit. You must insert code within EXIT 36 to create/populate these variables, then use VPUT to place the variables into the shared pool. Contact ASG Customer Support for additional information.
CUA	Displays the SmartEdit Action bar when set to ON. The default is OFF.
DB2	Allows SmartEdit to access the DB2 System Catalog when set to ON. The default is OFF.

Field	Description
IDMS	Allows SmartEdit to access the IDMS Data Dictionary when set to ON. The default is OFF.
LEARN	When LEARN is ON, the primary command that produces the same results as actions you have initiated from the pull-downs and pop-ups displays on the Generated Command pop-up. This is helpful if you want to learn the SmartEdit primary command syntax. For more information about the Generated Command pop-up see "Commands" on page 181 . The default for this Option is OFF.
LOCAL	Determines whether IDMS is accessed in Local Mode. The default is OFF.
XMODE	When XMODE is ON, excludes all lines from the screen before a primary command or a SmartEdit function is executed. Only the lines resulting from the command or function are displayed. The default is OFF.
SET	Specifies the current setting for each mode displays. You can change settings by typing over them.
Description	Provides descriptions of the options.

Usage Notes

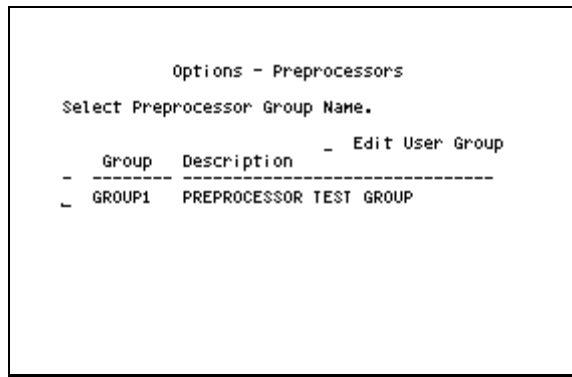
If you are using ISPF 4.1 through 4.8, these considerations apply:

- You can switch between the SmartEdit action bar and the ISPF Editor action bar by using the ISPF command SWITCH. This acts as a toggle to switch to the other action bar.
- If the SmartEdit action bar is removed using CUA OFF, the primary command SWITCH displays the ISPF Editor action bar and sets CUA ON.

Options - Preprocessors Pop-up

Use the Options - Preprocessors pop-up, shown in [Figure 124](#), to select a preprocessor group name.

Figure 124 • Options - Preprocessors Pop-up



Fields

Field	Description
Edit User Group	Allows editing of Preprocessors in list. Type any non-blank character to access the Options - Define Preprocessor Group pop-up.
Line Commands	Selects the preprocessor group name. Type any non-blank character to select the adjacent preprocessor group.

Options - Define Preprocessor Group Pop-up

Use the Options - Define Preprocessor Group pop-up, shown in [Figure 125](#), to devise a unique set of preprocessor steps. The group that you define displays on the Options - Preprocessor pop-up with a reserved description of USER GRP.

Figure 125 • Options - Define Preprocessor Group

```

Options - Define Preprocessor Group

Select the preprocessor for each step. Specify the
CLIST Name and Return Code for each USER step.

Description . . . USER DEFINED ENTRY

Step Types:
1. CICS only      4. DB2 - STDSQLNO  7. IDMS-Central
2. DLI only       5. DB2 - STDSQL86  8. User
3. CICS & DLI     6. IDMS - Local

Step  Type  CLIST Name  Max RC  Description
-  -  -  -  -
A    -    -          -    -
B    -    -          -    -
C    -    -          -    -
D    -    -          -    -
E    -    -          -    -
F    -    -          -    -
G    -    -          -    -

```

Fields

Field	Description
Description	Specifies a user-defined description of the preprocessor group.
Step Types	Specifies the step type. Type a number corresponding with the desired list selection from the Step Types displayed: <ul style="list-style-type: none"> 1. CICS only 2. DLI only 3. CICS & DLI 4. DB2 - STDSQLNO 5. DB2 - STDSQL86 6. IDMS - Local 7. IDMS - Central 8. User
CLIST Name	Specifies the CLIST name for the step.

Field	Description
Max RC	Specifies the maximum allowed Return Code for the CLIST step.
Description	Specifies a description of the CLIST step.

Options - Equate Pop-up

To define a name for a character string, follow this step:

- Select Options ► Equate. The Options - Equate pop-up, shown in [Figure 126](#), displays.

Figure 126 • Options - Equate Pop-up

Fields

Field	Description
Equate Name	<p>Specifies the name you want to assign to the character string. Names must conform to these standards:</p> <ul style="list-style-type: none"> • 1 to 10 alphanumeric characters • First character must be alphabetic • A hyphen is the only special character allowed • May be 1 to 4 DBCS characters
Equated String	<p>Specifies a character string to be substituted by the equate. A character string can be a long command, pattern, dataname, concatenated dataname, etc. You can also specify literals and blanks in the character string. Enclose character strings that include blanks with single or double quotes. The string may be a DBCS string.</p> <p>If the Equated String is left blank, the Equate Name is deleted.</p>

This chapter describes the Help pull-down options and contains these sections:

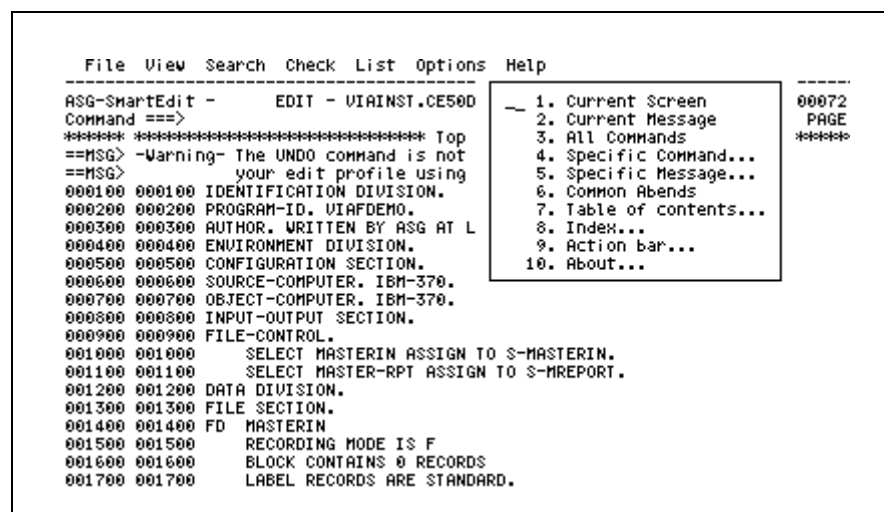
Section	Page
Help Pull-down	175
Help - Specific ASG Command Pop-up	177
Help - Specific ASG Message Number Pop-up	178
Help - About Pop-up	179

Help Pull-down

To access the online help facility, follow this step:

- Select Help on the action bar to display the Help pull-down shown in [Figure 127](#).

Figure 127 • Help Pull-down



Actions

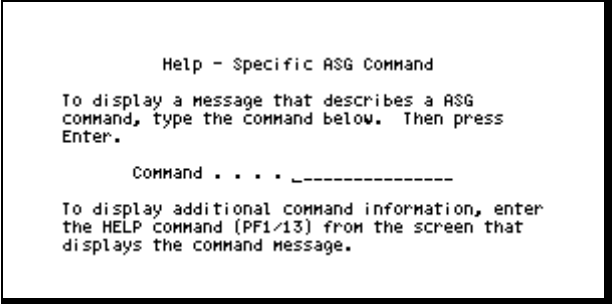
Action	Description
Current Screen	Displays help for the screen or pop-up that is currently displayed.
Current Message	Displays help for the message that is currently displayed.
All Commands	Displays a complete list of all SmartEdit primary commands, from which you can display information about a specific command by selecting the appropriate number.
Specific Command	Displays the Help - Specific ASG Command pop-up used to obtain help about a specific SmartEdit primary command.
Specific Message	Displays the Help - Specific ASG Message Number pop-up used to obtain help about a specific message number.
Common Abends	Displays the Abends screen, from which you can display information about a specific abend by selecting the appropriate number. Select number 2 on this screen to display the ASG Abend Codes screen, which lists all the ESW user abends and their explanations.
Table of contents	Displays the Help Table of Contents used to request help for general information.
Index	Displays the Help Index used to request help for specific information.
Action bar	Displays the Action Bar help screen that contains general information about the action bar and selections to help for each of the pull-downs.
About	Displays the Help - About pop-up that lists information about the currently installed levels of SmartEdit and Center.

Help - Specific ASG Command Pop-up

To obtain help for a specific SmartEdit primary command, follow this step:

- ▶ Select Help ▶ Specific Command. The Help - Specific ASG Command pop-up, shown in [Figure 128](#), displays.

Figure 128 • Help - Specific ASG Command Pop-up



Field

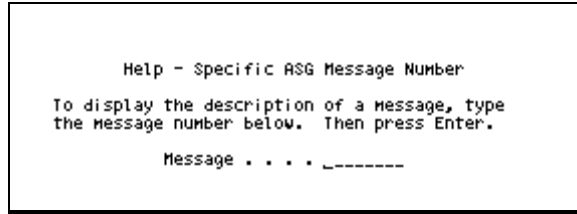
Field	Description
Command	Specifies the SmartEdit primary command for which to display the corresponding Help Tutorial. This field is required.

Help - Specific ASG Message Number Pop-up

To obtain help for a specific ASG message, follow this step:

- ▶ Select Help ▶ Specific Message. The Help - Specific ASG Message Number pop-up, shown in [Figure 129](#), displays.

Figure 129 • Help - Specific ASG Message Number Pop-up



```
Help - Specific ASG Message Number

To display the description of a message, type
the message number below. Then press Enter.

Message . . . . _____
```

Field

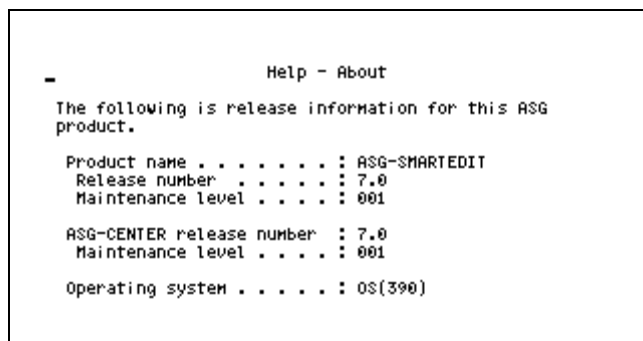
Field	Description
Message	Specifies the message number for which to display the HELP Explanation and Action screen. This field is required.

Help - About Pop-up

To displays information about the currently installed releases of SmartEdit and Center, follow this step:

- Select Help ► About. The Help - About pop-up, shown in [Figure 130](#), displays.

Figure 130 • Help - About Pop-up



```

-                               Help - About
The following is release information for this ASG
product.
Product name . . . . . : ASG-SMARTEDIT
Release number . . . . . : 7.0
Maintenance level . . . . . : 001
ASG-CENTER release number : 7.0
Maintenance level . . . . . : 001
Operating system . . . . . : OS(390)
  
```

Fields

Field	Description
Product name	Displays ASG-SmartEdit, or the name of the ESW product you are currently running, as well as the currently installed release number and maintenance level of SmartEdit
ASG-Center release number	Indicates the currently installed release number and maintenance level of Center.
Operating system	Specifies the operating system under that you are running.

11

Commands

This chapter describes the SmartEdit commands and contains these sections:

Section	Page
Introduction	183
Command Processing	184
Recalling/Repeating Commands	184
Cursor Position	185
Cursor Substitution Character	186
Command Diagrams	186
& (Retain) Command	189
ACTION Command	190
ALLOCDEF Command	191
BRANCH Command	192
CHECK Command	194
COBEDIT Command	197
CUA Command	198
DATAMAP Command	199
DISPLAY Command	200
EDITCOPY Command	201
EDITPGM Command	203

Section	Page
EQUATE Command	204
EXCLUDE Command	206
FINDXTND Command	212
HELP Command	222
HIGH Command	223
JUMP Command	229
LEVELS Command	230
LIST Command	231
LPRINT Command	233
LPUNCH Command	240
PARMDEF Command	246
PREF Command	247
PREPROC Command	250
PRINTLOG Command	251
PRINTLST Command	252
PRODLVL Command	253
RECALL Command	254
REDO Command	257
REFRESH Command	258
REPEAT Command	259
RESET Command	260
RFIND Command	261
RHIGH Command	262

Section	Page
RPREF Command	263
RSCROLL Command	264
RTREEVW Command	265
SCROLL Command	266
SET Command	272
SWITCH Command	274
TREEVIEW Command	275
UPDATE Command	277
ZOOM COPY Command	279
ZOOM DEF Command	280
ZOOMIN/ZOOMOUT Commands	281
Line Commands	283

Introduction

SmartEdit accepts primary and line commands entered in the same manner as ISPF commands. Primary commands are entered on screens and pop-ups that contain a command input area. Line commands are entered in the prefix area over the line numbers. SmartEdit supports all ISPF system commands on the appropriate screens. All SmartEdit commands are described in this chapter.

Command Processing

SmartEdit line commands are processed before primary commands. Commands entered by pressing a PF key are handled as if entered in the command input area. If a command is entered by pressing a PF key and there is a command in the command input area, the contents of the command input area are appended to the PF key command. The combined commands are then executed as a whole. Multiple commands entered in the command input area are separated by a semicolon (;) and are processed in a left-to-right sequence. Command results are displayed when one of these conditions occurs:

- The last command is processed and all commands in the sequence completed successfully.
- A command is not successfully processed. If an error occurs in the sequence, processing stops and an error message displays. Results are displayed for the commands that are successfully completed. The command causing the error and all remaining commands in the sequence are displayed in the command input area.

Each command in a command sequence is recognized separately for use with the RECALL command.

Recalling/Repeating Commands

SmartEdit remembers the last twenty primary commands entered. These methods may be used to repeat them:

- Type RECALL to display the last command entered in the command input area. Once the recalled command displays, you can modify, delete, or execute it again. Type RECALL repeatedly to display the commands in reverse sequence without executing them. To execute a recalled command, press Enter while the command displays in the command input area.
- Type REPEAT in the command input area to execute the last stacked primary command again (if it is a repeatable command).

These primary commands are not repeatable:

ACTION	PARMDEF	REPEAT
COBEDIT	PRINTLOG	RFIND
CUA	PRINTLST	RHIGH
HELP	PRODLVL	RSCROLL
JUMP	RECALL	RTREEVW
LEVELS	REDO	UPDATE

These commands repeat the last primary command:

Command	Description
& (retain)	If preceding a primary command, keeps the command displayed in the command input area after execution. This is the quickest way to repeat a command when minor changes are required before execution.
REDO	Reexecutes the last FINDXTND, HIGH, PREF, SCROLL, or TREEVIEW command (see the "REDO Command" on page 257 for more information).
RFIND	Repeats the last FINDXTND or ISPF FIND command from the current cursor location.
RHIGH	Repeats the last HIGH command from the current cursor location.
RPREF	Displays the last View - Paragraph Cross Reference pop-up, or functions as a PREF command with the cursor location as the target and default direction (see the "RPREF Command" on page 263 for more information).
RSCROLL	Repeats the last SCROLL command from the current cursor location.
RTREEVW	Redisplays the last Tree View screen.

Cursor Position

The result of the command depends on the starting point of the search. Specify a starting point by typing a line number or label name, or by using the cursor as the starting point. If the cursor is used as the starting point, these rules apply:

If cursor is in	Start the search
Command input area	in the first column of the first source line on the screen
Line command area	from the first column on that line
Program source	from the cursor position

Cursor Substitution Character

The cursor character is a token substitution character that can be used in any primary command on the editor screen. A token is a contiguous set of characters preceded by and followed by a blank, period, comma, or parenthesis. The cursor substitution character is defined on the Options - Product Parameters pop-up (see [Figure 114 on page 157](#) for more information). You can choose any character, except an underscore, but it should be one that is unique and rarely used in commands. The default is percent (%).

The cursor substitution character saves time and typing in commands. The cursor substitution character is typed anywhere on the command line, then the screen cursor is placed on the token that is to take its place in the command. For example, to find all occurrences of HLD-ZIP-PREFIX, type this FINDXTND command in the command input area:

```
FX %
```

Place the screen cursor anywhere on HLD-ZIP-PREFIX in the source code. This command is executed when you press Enter:

```
FX HLD-ZIP-PREFIX
```

Note that the cursor substitution character must have a space before and after it.

You can use multiple cursor tokens to specify consecutive tokens in the source code. For example, type this command in the command input area:

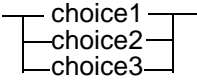
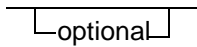
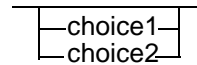
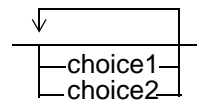
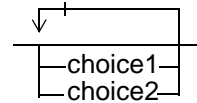
```
FX % % %
```

If the screen cursor is then placed on HLD-ZIP-PREFIX, the next two tokens are also picked up in the command, resulting in the command FX HLD-ZIP-PREFIX OF HLD-ZIP. The first token is HLD-ZIP-PREFIX, the second token is OF, and the third token is HLD-ZIP. The tokens must be consecutive.

Command Diagrams

These notational conventions and symbols are used to describe command syntax:

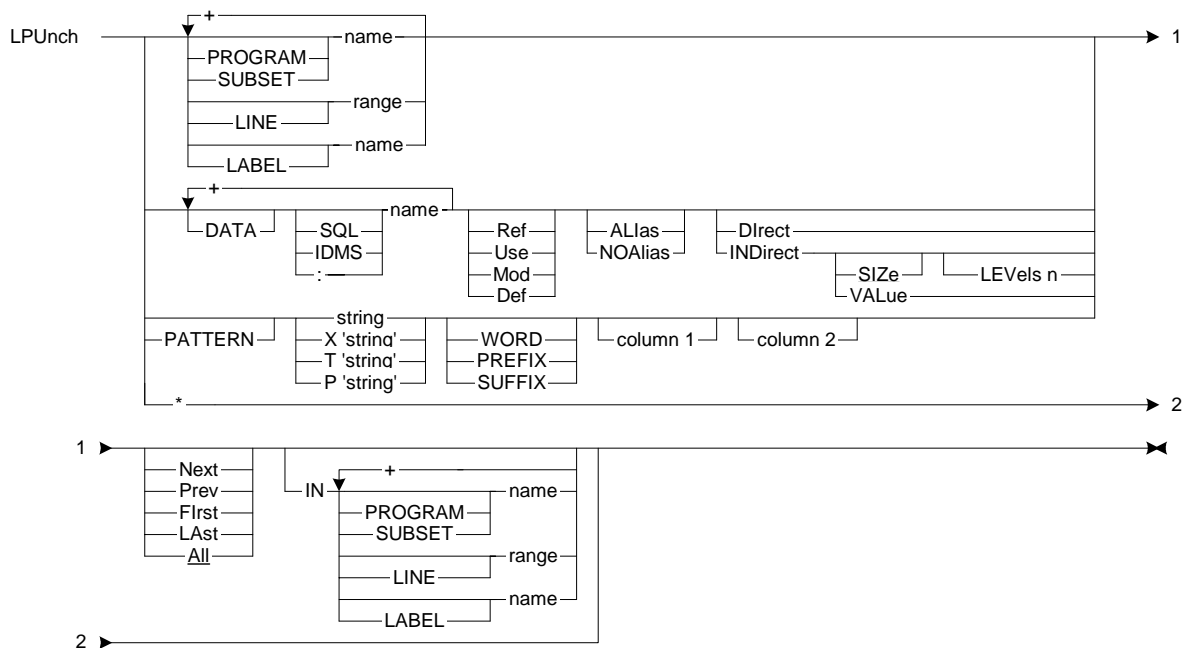
Item	Description
ABBREVIations	Illustrates the command abbreviation, which is shown in uppercase letters. Lowercase letters in the command are optional.
lowercase	Indicates user-supplied variable information.

Item	Description
UPPERCASE	Indicates commands or keywords.
Bold	Indicates operands that are available only if SmartEdit is installed and a SmartEdit analysis has been run on the COBOL program being tested.
Underline	Specifies the default value of an operand.
	Separates synonymous commands or operands.
—————→	Indicates that the command syntax is continued on the next line.
→—————	Indicates the command syntax is continued from the previous line.
—————✕	Indicates the end of the command syntax.
— required —	Indicates that the operand or keyword appearing on the main command line is required.
	Indicates that one operand is required.
	Indicates that an operand or keyword appearing below the main command line is optional.
	Indicates that operands are optional.
	Indicates that more than one operand can be chosen.
	Indicates that operands can be concatenated by placing a plus sign (+) between them.

The syntax diagram for the LPUNCH command is shown in [Figure 131](#). You must type LPUNCH with a target operand as indicated by the name variable on the main path of the line. If the path is followed down the first or second vertical line, you can specify any of the target operands or an asterisk (*). If you specify more than one operand, separate each with a space. Operands that can be concatenated are indicated by a returning arrow that includes a + in the line. Dataname operands (i.e., RER, USE) pertain to all datanames in a concatenated series.


Continuation lines are numbered 1 and 2, with 1 being the main path of the line. Continuation line 2 shows that the * operand is entered with no other values since the line from it extends to the end of the path, bypassing the direction and intarget operands. The direction and intarget operands are optional since they are below the main path of the line.

Figure 131 • Command Syntax Diagram



& (Retain) Command

&any primary command



Function

Executes the specified primary command and keeps it displayed in the command input area for repeated use or modification.

Operands

The & must be followed by a primary command.

Usage Notes

The retain command is useful if the same primary command is to be executed repeatedly, or if minor changes to a command are desired. For example, after executing the FINDXTND command for a data item, it may be desirable to execute an LPRINT command (see ["LPRINT Command" on page 233](#) for more information) on the same data item. The retain command performs this function quickly and easily.

Example

To find the EOF-FLAG and keep &FINDXTND EOF-FLAG displayed in the command input area, type this command:

```
&FINDXTND EOF-FLAG
```

ACTION Command

ACTion 

Function

Initiates the cursor-sensitive function previously selected from a pull-down or pop-up.

Operands

None.

Usage Notes

Use the ACTION command when an option or action is selected on a pull-down or pop-up that requires use of the cursor position as the starting point for the action. When you press Enter on the pull-down or pop-up, the editor screen is redisplayed. The ACTION command displays in the command input area and a message displays reminding you to position the cursor before pressing Enter.

ALLOCDEF Command

ALLOCDEF | ADEF

Function

Displays the Options - Product Allocations pop-up (shown in [Figure 120 on page 164](#)), which is used to specify the DASD volumes for the Log, List, Punch, and Work files. You can also use this command to specify space for the Work file.

Operands

None.

Usage Notes

You can also display the Options - Product Allocations pop-up by selecting Options ► Product Allocations.

BRANCH Command



Function

Positions the cursor at the specified target. Use this command to scroll from a statement, such as a PERFORM, to the paragraph being performed. Use the BACKUP operand to return to the statement from which the branch occurred.

Operands

Operand	Description
Blank	Indicates that the BRANCH command will automatically locate the target of the GO TO, PERFORM, or CALL statement. No operand is required if the cursor is positioned on a GO TO, PERFORM, or CALL statement. If the cursor is positioned on any other statement, the BRANCH command locates the next logical statement to be executed. The PROCEDURE DIVISION label is located if the cursor is positioned outside the PROCEDURE DIVISION.
LABEL <i>name</i>	Locates the specified paragraph or section name. If the specified name is not fully qualified and there are multiple paragraphs or sections with the specified name, the cursor is positioned on the first paragraph or section found. A message displays indicating the number of paragraphs or sections found.
PROGRAM <i>name</i>	Locates the specified nested program. For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times by using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, then it can be qualified as P120-READ OF VIAFDEM1.
BACKUp	Positions the cursor at the location from which the branch occurred.

Usage Notes

You can also initiate the Branch function by selecting Search ► Branch, to display the Search - Branch Request pop-up. See ["Search" on page 107](#) for more information.

The BRANCH command is helpful when tracking branching logic. It enables you to track branching logic several levels deep into PERFORMed code, then return to each PERFORM statement.

An effective way to use the BRANCH command is to type the command in the command input area, position the cursor on a PERFORM, GO TO, or CALL statement, and press Enter. The screen position is saved for use with the BACKUP operand based on these conditions:

- A LABEL name is typed on the command line.
- The cursor is placed on a GOTO or PERFORM statement.

The PROCEDURE DIVISION label displays when the BRANCH command is used with no operands and the cursor is positioned in a part of the program other than the PROCEDURE DIVISION.

If a PROGRAM EXIT or STOP RUN is encountered, the LOGICAL END OF PROGRAM short message displays indicating that the logical end of the program has been reached. If the end of the source file is reached, the PHYSICAL END OF PROGRAM short message displays indicating that you cannot branch any further.

You can also use the BR (Branch) line command when tracking branching logic.

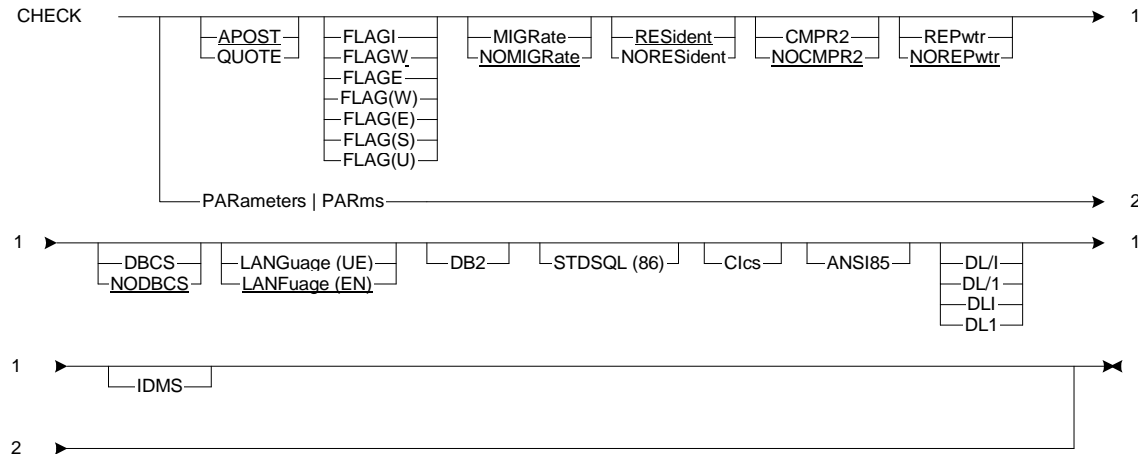
Examples

Type BRANCH, place the cursor on PERFORM READ-ALL, and press Enter. The screen is scrolled and the cursor is positioned on the READ-ALL paragraph.

Type B P120-READ. The screen is scrolled and the cursor is positioned on the P120-READ paragraph.

Type B PROGRAM-INIT. The screen is scrolled and the cursor is positioned on the PROGRAM-INIT paragraph. Then type B BACKUP. The screen is scrolled and the cursor is positioned at the location from where you entered the B PROGRAM-INIT command.

CHECK Command



Function

Provides an online syntax check mechanism for the COBOL source currently being edited. The CHECK command performs a syntax check of the program and places syntax error messages directly below the line in which any errors are found.

Operands

Note:

The operands of the CHECK command temporarily override the values specified on the Check - Parameters pop-up.

Operand	Description
Blank	Specifies that the defaults on the Check - Parameters pop-up will be used.
APOST	Specifies the compiler default for COBOL II and COBOL/370. The language level is specified on the Options - COPY/Include Libraries screen (shown in Figure 11 on page 24).
QUOTE	Specifies the compiler default for COBOL II. The language level is specified on the Options - COPY/Include Libraries screen.
FLAGI	Specifies that all informational messages are produced.
FLAGW or FLAG(W)	Specifies that all messages are produced.

Operand	Description
FLAGE or FLAG(E)	Specifies that only error messages are produced. Warning messages and other diagnostics are suppressed.
FLAG(S)	Specifies that only severe error messages are produced. This operand is only available for COBOL II and COBOL/370.
FLAG(U)	Specifies that only unrecoverable error messages are produced. This operand is only available for COBOL II and COBOL/370.
MIGRate	Flags statements that require changes in order to function in the next level of the COBOL compiler. This operand is only available for COBOL II and COBOL/370.
NOMIGRate	Specifies that statements that require changes to function in the next level of the COBOL compiler are not flagged. NOMIGRATE is the default.
RESident	Sets the compiler flag to RES. This is the default. This operand is only available for COBOL II and COBOL/370.
NORESident	Sets the compiler flag is set to NORES.
DBCS	Sets the compiler DBCS flag.
NODBCS	Specifies that the compiler DBCS flag will not be set. This is the default.
CMPR2	Specifies that the COBOL II Release 3 compiler functions the same as the COBOL II Release 2 compiler. This operand is only available for COBOL II and COBOL/370.
NOCMPR2	Specifies that the COBOL II Release 3 compiler does not function the same as the COBOL II Release 2 compiler. This is the default.
REPwtr	Invokes the IBM COBOL Report Writer Precompiler.
NOREPwtr	Specifies that no report writer precompiler is invoked. This is the default.
LANGuage(UE)	Sets the UPPERCASE ENGLISH flag. This operand is only available for COBOL II and COBOL/370.
LANGuage(EN)	Sets the ENGLISH flag. This operand is only available for COBOL II and COBOL/370.
DB2	Specifies that the program to be checked contains DB2 command level statements.

Operand	Description
STDSQL(86)	Specifies that SQL is ANSI 86 standard SQL.
Cics	Specifies that the program to be checked contains CICS command level statements.
ANSI85	Calls the CICS preprocessor with the ANSI85 option.
DLI DL1 DL/I DL/1	Specifies that the program to be checked contains DL1 command level statements.
IDMS	Specifies that the program to be checked contains IDMS command level statements.
PARameters PARms	Displays the Check - Parameters pop-up used to set parameters for the CHECK command.

Usage Notes

You can also initiate an online syntax check by selecting Perform Syntax Check on the Check pull-down. See ["Check" on page 137](#) for more information.

A message displays stating the number of error(s) encountered. To return the source file to its original unmarked form, use the RESET command.

The compiler dataset name is specified at the time that SmartEdit is installed, and can be changed on the Check - Parameters pop-up. Copy libraries can be changed on the Options - COPY/Include Libraries screen (see [Figure 11 on page 24](#) for more information).

To view the errors, use the SCROLL TAG, SCROLL HI, and RSCROLL commands.

The CHECK command uses temporary files for its processing. If CHECK fails due to insufficient temporary file space, then increase the size of the Work file on the Options - Product Allocations pop-up. Select Options ► Product Allocations to display the Options - Product Allocations pop-up.

COBEDIT Command



Function

Activates and deactivates SmartEdit under ISPF/PDF Edit.

Operands

Operand	Description
Blank	Specifies that when you enter the COBEDIT command with no operands, the default of ON is used.
ON	Activates SmartEdit. All SmartEdit commands are available for use.
OFF	Deactivates SmartEdit. SmartEdit commands are no longer available. The user is returned to the ISPF/PDF editor.

Usage Notes

The COBEDIT command can only be typed on the ISPF/PDF Edit screen.

The COBEDIT command may not be needed to activate a SmartEdit session, depending on the profile of the dataset being edited.

CUA Command



Function

Displays or removes the SmartEdit CUA action bar from the screen for the current session. To make the setting permanent for your profile, use the SET command (e.g., SET CUA ON) to display the CUA action.

Operands

Operand	Description
Blank	Assumes the default operand ON and displays the action bar if it is not already displayed.
ON	Displays the SmartEdit action bar and activates the CUA interface for SmartEdit. This is the default.
OFF	Removes the action bar from the screen and deactivates the CUA interface for SmartEdit.

Usage Notes

If you are using ISPF 4.1 through 4.8, these considerations apply:

- You can switch between the SmartEdit action bar and the ISPF Editor action bar by using the primary command SWITCH. This acts as a toggle to switch to the other action bar.
- If the SmartEdit action bar is removed using CUA OFF, the primary command SWITCH displays the ISPF Editor action bar and sets CUA ON.

DATAMAP Command

DATAmop 

Function

Use the DATAMAP (DM) command to display the length of a data item and the length and offset for any data items that are contained within the selected data item. You can also use the DM line command to display the length of a data item.

Operands

Allows one data variable name operand.

Usage Notes

The DATAMAP command is used in conjunction with the cursor location. You can use this command in any of these ways:

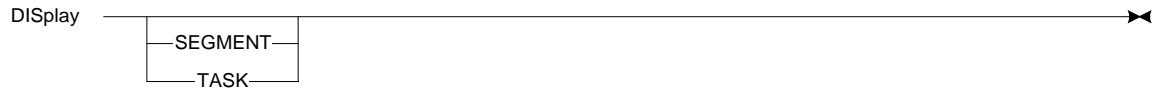
- Type DATA (or DM) in the command input area, position the cursor to the line containing the target data item, and press Enter.
- Type DATA (or DM) plus the name of the data item on the command line and press Enter. For example:

DATA REC-CNT

- Type DM in the line command area to the left of the data item and press Enter.

Data items are displayed on the Data Item Offset and Length pop-up. See ["View" on page 87](#) for more information.

DISPLAY Command



Function

Displays all statements contained in the active Logic Segment (created in Encore), or statements found using the Task facility in Insight. The DISPLAY command is only valid if Encore and/or Insight are installed.

Operands

Operand	Description
Blank	Defaults to the operand corresponding to the ESW product from which you entered SmartEdit.
SEGMENT	<p>Excludes from the screen all statements in the program that are not contained in the active Logic Segment created in Encore. This operand is only valid if Encore is installed.</p> <p>If there is no active Logic Segment, the File - Display Logic Segment Entry pop-up displays to specify a Logic Segment. See "File" on page 77 for more information.</p> <p>This is the default if SmartEdit is the only active ESW product, or if SmartEdit was entered through Encore.</p>
TASK	Displays statements that were marked while using the Task facility in Insight. This operand is only valid if Insight is installed.

Statements must be marked in the Insight Task facility before entering this operand. This is the default if SmartEdit was entered through Insight.

Usage Notes

You can also display statements by selecting File ► Display Segment or File ► Display Task. See ["File" on page 77](#) for more information.

When using the DISPLAY command for a tasked editing session from Insight, even when XMODE equals OFF, only highlighted lines are selected for display in SmartEdit.

EDITCOPY Command



Function

Locates a copybook and opens it in an edit session using your default editor. If SmartEdit is not your default editor, an ISPF edit session is opened for the specified copybook. You can use this command on all copybooks (except certain Includes that you can zoom in on) that are currently viewable in SmartEdit.

The EDITCOPY (EC) command applies to all copybooks that exist in the Copy/Include libraries you specified on the Options - COPY/Include Libraries screen in SmartEdit.

Operands

Operand	Description
PAN	Specifies a search only in Panvalet libraries.
LIB	Specifies a search only in Librarian libraries.
PDS	Specifies a search only in partitioned datasets.
<i>copybookname</i>	Specifies the name of the copybook you want to edit.

Usage Notes

The EDITCOPY command is used in conjunction with the cursor location. You can use this command in any of these ways:

- Type EDITC (or EC) in the command input area, position the cursor on the name of the copybook, and press Enter.
- Type EDITC (or EC) plus the name of the copybook on the command line, and press Enter. For example:


```
EDITC copybookname
```

- Type EC in the line command area to the left of the copybook name, place the cursor on the name of the copybook, and press Enter.

These conditions apply for the EDITCOPY command:

- Panvalet does not allow recursive edits. If you use EDITCOPY in Panvalet, SmartEdit displays a message indicating the location of the copybook, the dataset source type, and the Source Manager type.
- If you use EDITCOPY in SCLM, an ISPF edit session is opened and the copybook is no longer under SCLM control.
- If you use EDITCOPY in Librarian, you must use the ELIPS Librarian Source Manager. If you do not, SmartEdit displays a message indicating the location of the copybook, the dataset source type, and the Source Manager type.
- The EDITCOPY command is not supported in other source manager editors.

EDITPGM Command

EDITPgm 

Function

Locates a CALLED program and opens it in SmartEdit. You can use this command on all programs that reside in the same library as the current member.

Operands

Operand	Description
<i>programname</i>	Specifies the name of the program you want to edit.

Usage Notes

The EDITPGM (EP) command is used in conjunction with the cursor location. You can use this command in any of these ways:

- Type EDITP (or EP) in the command input area, position the cursor on the name of the program, and press Enter.
- Type EDITP (or EP) plus the program name on the command line, and press Enter. For example:

EDITP *programname*

- Type EP in the line command area to the left of the program name, place the cursor on the name of the program, and press Enter.

EQUATE Command

Equate 

Function

Defines a name for a character string.

Operands

Operand	Description
<i>name</i>	<p>Specifies the name of the character string. Names must conform to these standards:</p> <ul style="list-style-type: none"> • 1 to 10 alphanumeric characters • First character must be alphabetic • A hyphen is the only special character allowed • May be 1 to 4 DBCS characters
<i>string</i>	<p>Specifies a character string to be substituted by the EQUATE command. A character string can be a long command, pattern, dataname, concatenated dataname, etc. You can also specify literals and blanks in the character string. Enclose character strings that include blanks in single or double quotes. The string may be a DBCS string.</p> <p>If you do not enter the string operand, the equated name is deleted.</p>

Usage Notes

Note: _____

You can also define and delete Equates using the Options - Equate pop-up. See ["Options" on page 155](#) for more information.

Use equated names during a SmartEdit session to reduce the number of keystrokes. You can use multiple equates, if desired.

If an equated name is used in another EQUATE command, it must first be defined. For example:

```
EQ CC TOT-COST
EQ BB CC + PROD-COST
```

BB now contains $TOT-COST + PROD-COST$.

To delete an equated name, type EQUATE and the name without the string.

Examples

To equate the name A to 'EOF-FLAG', type this command:

```
EQ A 'EOF-FLAG'
```

Then you can enter this command to find 'EOF-FLAG':

```
FX A
```

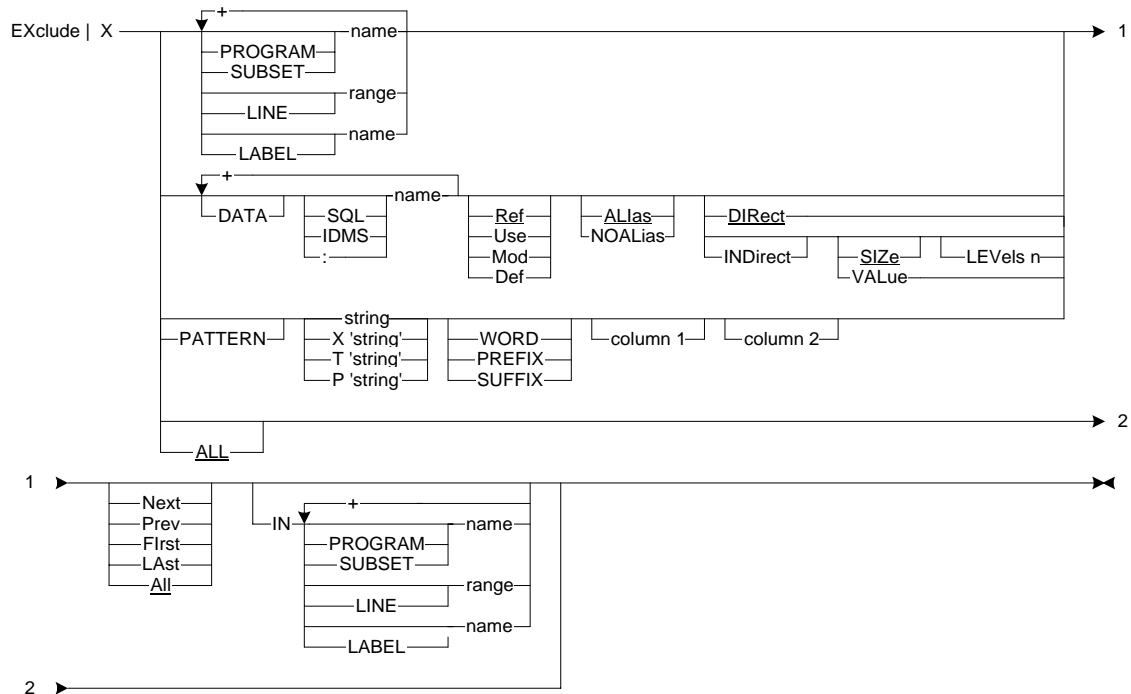
To equate AA to TOT-COST, type this command:

```
EQ AA TOT-COST
```

Then you can enter this command to change AA to be $TOT-COST * PROD-COST$:

```
EQ AA AA * PROD-COST
```

EXCLUDE Command



Function

Performs a FINDXTND command on the specified target, excluding the resulting lines.

Note:

To use the ISPF/PDF version of the EXCLUDE command instead of the SmartEdit version, set the Editor-Exclude-Command parameter to YES in VIA\$PRME and re-execute VIA\$PRME. See the *ASG-Center Installation Guide* for more information.

Operands

Operand	Description																																										
PROGRAM <i>name</i>	Specifies the name of the main program or any nested program representing all the code contained in the program. This includes all the programs physically nested inside the specified program.																																										
SUBSET <i>name</i>	<p>Specifies a predefined COBOL language subset. These are the predefined COBOL language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CALL</td><td>DIRECTive</td><td>LABel</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTion</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table> <p>A screen subset:</p> <table><tr><td>Highlighted</td><td> </td><td>HI</td></tr><tr><td>NONHighlighted</td><td> </td><td>NHI</td></tr><tr><td>Excluded</td><td> </td><td>X</td></tr><tr><td>NONExcluded</td><td> </td><td>NX</td></tr></table> <p>A tagged lines subset of tags displaying in columns 73 through 80.</p> <p>See "Product Overview" on page 5 for a description of each subset.</p>	ASsignment	DEFinition	IO	CALL	DIRECTive	LABel	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTion	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS		Highlighted		HI	NONHighlighted		NHI	Excluded		X	NONExcluded		NX
ASsignment	DEFinition	IO																																									
CALL	DIRECTive	LABel																																									
CIcs	DIVision	MATH																																									
COBOLII	DL/I	Output																																									
COBOL/370	DML	PARagraph																																									
COMment	ENtry	PERform																																									
CONditional	EXIt	SECTion																																									
DB2 SQL	GOto	SORTMerge																																									
DDL	IDMS	STructure																																									
DEBug	IMS																																										
Highlighted		HI																																									
NONHighlighted		NHI																																									
Excluded		X																																									
NONExcluded		NX																																									
LINE <i>range</i>	Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor screen.																																										
LABEL <i>name</i>	Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.																																										
DATA <i>name</i>	<p>Specifies a COBOL dataname or qualified COBOL dataname. Dataname refers to any valid COBOL reference for a data element.</p> <p>These subordinate operands apply to the dataname and can be used to qualify the EXCLUDE command: REF, USE, MOD, DEF, ALIAS, NOALIAS, DIRECT, and INDIRECT. The defaults are REF, ALIAS, and DIRECT.</p>																																										
SQL <i>name</i>	Excludes datanames that are DB2/SQL variables only.																																										
IDMS <i>name</i>	Excludes datanames that are IDMS variables only.																																										

Operand	Description
: <i>name</i>	Excludes datanames that are COBOL variables only.
Ref	Excludes occurrences of the dataname where its value is defined, tested, used, set, or modified. This is the default value for the DATA name operand.
Use	Excludes occurrences of the dataname where its value is being tested or used.
Mod	Excludes occurrences of the dataname where its value is being set or modified.
Def	Excludes definitions of the dataname in the DATA DIVISION.
ALias	Excludes occurrences of the aliases for the dataname. These aliases include: <ul style="list-style-type: none"> • Parent - higher level group item • Child - lower level item • Rename/Redefinition - renamed, redefined, or 88 level items This is the default value for the DATA name operand.
NOAlias	Ignores aliases for the specified dataname.
DIRECT	Excludes occurrences of the dataname (and aliases if specified) where it is directly tested, used, set, modified, or defined (based on the REF, USE, MOD, or DEF selection). The default for the DATA name operand is direct; however, if the SIZE, VALUE, or LEVELS operand is specified, INDIRECT is assumed.
INDirect	Excludes any dataname indirectly affected by the specified dataname (and aliases if specified). You can further qualify the indirect data item using the SIZE, VALUE, and LEVELS subordinate operands. These subordinate operands indicate the type of indirect reference to be excluded. SIZE and ALL LEVELS are the defaults for the INDIRECT operand.
SIZE	Excludes datanames that could be directly or indirectly affected if the size of the specified dataname were to be changed. SIZE is only valid with the INDIRECT operand. This is the default for the INDIRECT operand.
VALue	Excludes occurrences of datanames that are directly or indirectly affected by a change in the value of the specified dataname. The LEVELS subordinate operand is not used with VALUE. VALUE is only valid with the INDIRECT operand.

Operand	Description
LEVelS <i>n</i>	Includes all data items that are affected within the specified number of indirect levels. The VALUE subordinate operand is not used with LEVELS. ALL LEVELS is the default for the LEVELS operand. LEVELS is only valid with the INDIRECT operand.
PATTERN	Specifies an optional keyword indicating the characters that follow are part of a string.
<i>string</i>	Specifies a string of alphanumeric or DBCS characters. If the pattern string contains blanks, it must be enclosed in single or double quotes. You can further qualify the pattern string using the WORD, PREFIX, or SUFFIX subordinate operands. These subordinate operands describe how the pattern string is to be used.
X' <i>string</i> '	Specifies the hexadecimal string option. Specific unprintable characters may be specified by giving the EBCDIC hexadecimal value. The string must be enclosed in single or double quotes.
T' <i>string</i> '	Specifies the text string option. A character string may be entered regardless of upper or lowercase by using the text option. The string must be enclosed in single or double quotes.
P' <i>string</i> '	Specifies the picture string option. You can enter a string profile instead of exact characters. Enclose the string in single or double quotes. Nine special characters are defined by SmartEdit for use in picture strings. You can also combine these special characters with other characters. These characters are: <p> P'=' Any character P'¬' Any nonblank character P'.' Any nondisplay character P'#' Any numeric character P'-' Any non-numeric character P'@' Any alphabetic character (upper or lowercase) P'<' Any lowercase alphabetic character P'>' Any uppercase alphabetic character P'\$' Any special character (not alphabetic or numeric) </p>
WORD	Specifies the specified pattern string preceded and followed by any non-alphanumeric character (except a hyphen).
PREFIX	Specifies a word that begins with the specified pattern string.
SUFFIX	Specifies a word that ends with the specified pattern string.

Operand	Description
Column1	Specifies the column number where the search is to begin.
Column2	Specifies the column number where the search is to end.
Blank	Excludes all lines from the display.
ALL	Excludes all lines from the display. The ALL operand cannot be entered with any other operands. ALL is the default if you enter EXCLUDE with no operands.
Next	Searches forward from the current cursor position to the next occurrence of the requested target.
Prev	Searches backward from the current cursor position to the previous occurrence of the requested target.
First	Searches from the top of the source file to the first occurrence of the requested target.
LAst	Searches backward from the bottom of the source file to the first occurrence of the requested target.
All	Searches for all occurrences of the requested target and displays the number found in the short/long message field. This is the default value for the EXCLUDE command.
IN	Restricts the EXCLUDE command to the specified target type. This operand is optional.

Usage Notes

You can also exclude lines using the X (Exclude) and XX (Exclude Block) line commands. Selecting View ► Exclude displays the View - Exclude Request pop-up, or you can use one of the Search pop-ups available on the Search pull-down.

See ["View" on page 87](#) for more information about the View-Exclude Request pop-ups and ["Search" on page 107](#) for more information about the Search pop-ups and Search pull-down.

The EXCLUDE command removes specific lines from the screen resulting from a previous command. Excluded lines are represented by a line of dashes and text stating *n* LINE(S) NOT DISPLAYED.

To concatenate targets, place a plus sign (+) between the target names. For example:

```
IO + CALL
```

These rules apply to concatenation:

- You can use a concatenated dataname wherever a dataname is valid. Dataname subordinate operands (REF, ALIAS, etc.) pertain to all datanames in a concatenated series.
- You can use a concatenated set name wherever a set name is valid. These sets are valid:

LABEL LINE PROGRAM SUBSET

For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times by using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, then you could qualify it as P120-READ OF VIAFDEM1.

Example

To remove all non-highlighted lines from the screen, type this screen:

EXCLUDE NHI

The EXCLUDE command supports the ISPF .label function with a target or an IN target.

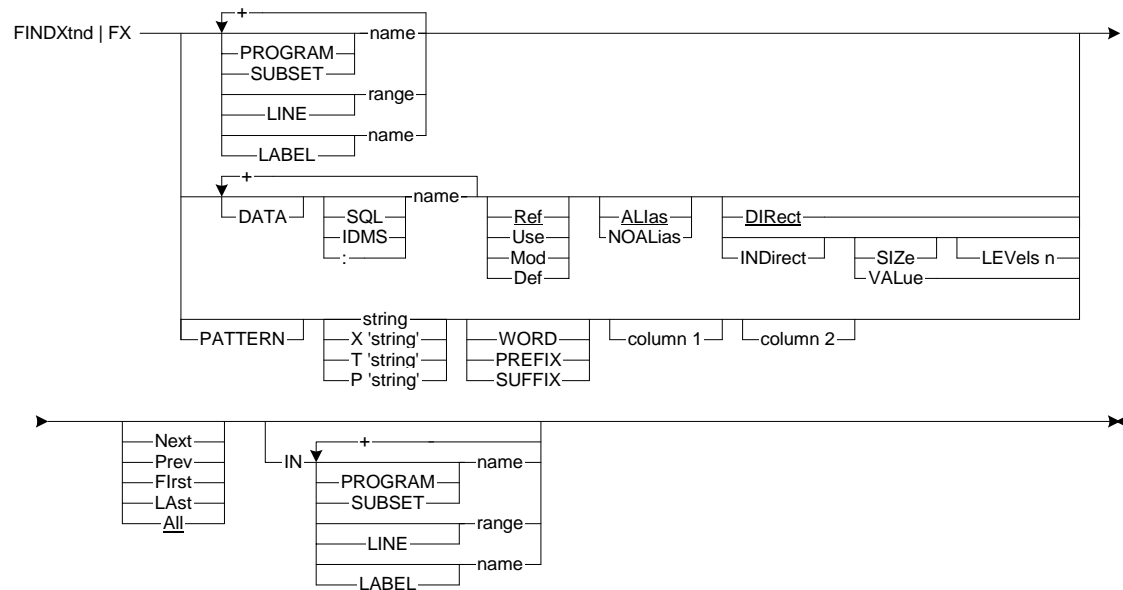
To use .label with a target, type this command:

EX .A

To use .label with an IN target (DATE-CODE), type this command:

EX DATE-CODE IN .A

FINDXTND Command



Function

The FINDXTND command performs a COBOL intelligent search of the source code for one or all occurrences of the specified target.

Operands

Operand	Description																																										
PROGRAM <i>name</i>	Specifies the name of the main program or any nested program representing all the code contained in the program. This includes all the programs physically nested inside the specified program.																																										
SUBSET <i>name</i>	<p>Specifies a predefined COBOL language subset. These are the predefined COBOL language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CALL</td><td>DIRECTive</td><td>LABel</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTion</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table> <p>A screen subset:</p> <table><tr><td>Highlighted</td><td> </td><td>HI</td></tr><tr><td>NONHighlighted</td><td> </td><td>NHI</td></tr><tr><td>Excluded</td><td> </td><td>X</td></tr><tr><td>NONExcluded</td><td> </td><td>NX</td></tr></table> <p>A tagged lines subset of tags displaying in columns 73 through 80.</p> <p>See "Product Overview" on page 5 for a description of each subset.</p>	ASsignment	DEFinition	IO	CALL	DIRECTive	LABel	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTion	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS		Highlighted		HI	NONHighlighted		NHI	Excluded		X	NONExcluded		NX
ASsignment	DEFinition	IO																																									
CALL	DIRECTive	LABel																																									
CIcs	DIVision	MATH																																									
COBOLII	DL/I	Output																																									
COBOL/370	DML	PARagraph																																									
COMment	ENtry	PERform																																									
CONditional	EXIt	SECTion																																									
DB2 SQL	GOto	SORTMerge																																									
DDL	IDMS	STructure																																									
DEBug	IMS																																										
Highlighted		HI																																									
NONHighlighted		NHI																																									
Excluded		X																																									
NONExcluded		NX																																									
LINE <i>range</i>	Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor screen.																																										
LABEL <i>name</i>	Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.																																										
DATA <i>name</i>	<p>Specifies a COBOL dataname or qualified COBOL dataname. Dataname refers to any valid COBOL reference for a data element.</p> <p>These subordinate operands apply to the dataname and can be used to qualify the EXCLUDE command: REF, USE, MOD, DEF, ALIAS, NOALIAS, DIRECT, and INDIRECT. The defaults are REF, ALIAS, and DIRECT.</p>																																										
SQL <i>name</i>	Includes datanames that are DB2/SQL variables only.																																										
IDMS <i>name</i>	Includes datanames that are IDMS variables only.																																										

Operand	Description
: <i>name</i>	Includes datanames that are COBOL variables only.
Ref	Includes occurrences of the dataname where its value is defined, tested, used, set, or modified. This is the default value for the DATA name operand.
Use	Includes occurrences of the dataname where its value is tested or used.
Mod	Includes occurrences of the dataname where its value is set or modified.
Def	Includes definitions of the dataname in the DATA DIVISION.
ALias	Includes aliases for the dataname. These aliases include: <ul style="list-style-type: none"> • Parent - higher level group item • Child - lower level item • Rename/Redefinition - renamed, redefined, or 88 level items This is the default value for the DATA name operand.
NOAlias	Ignores aliases for the specified dataname.
DIREct	Includes occurrences of the dataname (and aliases if specified) where it is directly tested, used, set, modified, or defined (based on the REF, USE, MOD, or DEF selection). The default for the DATA name operand is direct; however, if the SIZE, VALUE, or LEVELS operand is specified, INDIRECT is assumed.
INDirect	Includes any dataname indirectly affected by the specified dataname (and aliases if specified). You can further qualify the indirect data item using the SIZE, VALUE, and LEVELS subordinate operands. These subordinate operands indicate the type of indirect reference to be located. SIZE and ALL LEVELS are the defaults for the INDIRECT operand.
SIZE	Includes datanames that could be directly or indirectly affected if the size of the specified dataname were to be changed. SIZE is only valid with the INDIRECT operand. This is the default for the INDIRECT operand.
VALue	Includes occurrences of datanames that are directly or indirectly affected by a change in the value of the specified dataname. The LEVELS subordinate operand is not used with VALUE. VALUE is only valid with the INDIRECT operand.

Operand	Description
LEVelS <i>n</i>	Includes all data items that are affected within the specified number of indirect levels. The VALUE subordinate operand is not used with LEVELS. ALL LEVELS is the default for the LEVELS operand. LEVELS is only valid with the INDIRECT operand.
PATTERN	Specifies an optional keyword indicating the characters that follow are part of a string.
<i>string</i>	Specifies a string of alphanumeric or DBCS characters. If the pattern string contains blanks or quotes, it must be enclosed in single or double quotes. You can further qualify the pattern string using the WORD, PREFIX, or SUFFIX subordinate operands. These subordinate operands describe how the pattern string is to be used.
X' <i>string</i> '	Specifies the hexadecimal string option. Specific unprintable characters may be specified by giving the EBCDIC hexadecimal value. The string must be enclosed in single or double quotes.
T' <i>string</i> '	Specifies the text string option. A character string may be entered regardless of upper or lowercase by using the text option. The string must be enclosed in single or double quotes.
P' <i>string</i> '	Specifies the picture string option. A string profile may be entered instead of exact characters. The string must be enclosed in single or double quotes. Nine special characters are defined by SmartEdit for use in picture strings. You can combine these special characters with other characters. These characters are: <p>P'=' Any character</p> <p>P'¬' Any nonblank character</p> <p>P'.' Any nondisplay character</p> <p>P'#' Any numeric character</p> <p>P'-' Any non-numeric character</p> <p>P'@' Any alphabetic character (upper or lowercase)</p> <p>P'<' Any lowercase alphabetic character</p> <p>P'>' Any uppercase alphabetic character</p> <p>P'\$' Any special character (not alphabetic or numeric)</p>
WORD	Specifies the specified pattern string preceded and followed by any non-alphanumeric character (except a hyphen).
PREFIX	Specifies a word that begins with the specified pattern string.
SUFFIX	Specifies a word that ends with the specified pattern string.

Operand	Description
Column1	Specifies the column number where the search is to begin.
Column2	Specifies the column number where the search is to end.
Blank	Excludes all lines from the display.
Next	Searches forward from the current cursor position to the next occurrence of the requested target.
Prev	Searches backward from the current cursor position to the previous occurrence of the requested target.
Ffirst	Searches from the top of the source file to the first occurrence of the requested target.
LAst	Searches backward from the bottom of the source file to the first occurrence of the requested target.
All	Searches for all occurrences of the requested target and displays the number found in the short/long message field. This is the default value for the FINDXTND command.
IN	Restricts the FINDXTND command to the specified target type. This operand is optional.

Usage Notes

A search can also be conducted by selecting a Search pop-up on the Search pull-down. See ["Search" on page 107](#) for more information.

For users of ISPF 4.1 through 4.8, make sure the ISPF setting, long message in pop-up, is disabled when executing the FX command. If you do not, a portion of the display screen is overlaid with the FX results formatted in a Long Message pop-up.

To concatenate targets, place a plus sign (+) between the target names, as shown in this example:

```
IO + CALL
```

These rules apply to concatenation:

- You can use a concatenated dataname wherever a dataname is valid. Dataname subordinate operands (REF, ALIAS, etc.) pertain to all datanames in a concatenated series.
- You can use a concatenated set name wherever a set name is valid. These sets are valid:

LABEL LINE PROGRAM SUBSET

For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times by using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, then you could qualify it as P120-READ OF VIAFDEM1.

If lines containing targets are excluded, they are redisplayed on the screen. Tags are placed on the source code lines in columns 1 through 6 of the editor screen, indicating the type of target found. These are the valid tags:

Tag	Description
LABEL <i>name</i>	LABEL RG
PROGRAM <i>name</i>	PROGRAM PGM tag
SUBSET <i>name</i> - COBOL	ASsignment DEFinition IO CAIl DIRective LABel CIcs DIVision MATH COBOLII DL/I Output COBOL/370 DML PARagraph COMment ENtry PERform CONditional EXIt SECTion DB2 SQL GOTO SORTMerge DDL IDMS STructure DEBug IMS
SUBSET <i>name</i> - SCREEN	HIGH NONHIGH EXCLUDE NONX
LINE <i>range</i>	LINE RNG
DATA <i>name</i>	REF USE MOD DEF

Tag	Description
PATTERN <i>string</i>	<p>Specifies that the NEXT and PREV operands start the search from the current position and locate the closest occurrence. If ALL is specified, all lines are searched regardless of the current line or direction. A message displays indicating the number of targets found.</p> <p>When an unqualified dataname is entered and more than one occurrence of the specified dataname exists, all occurrences are found. This provides a means of seeing all like datanames but under different group level names, then selecting the appropriate dataname. A message displays if an invalid qualification is entered.</p> <p>When a dataname is entered with the REF operand, the screen is positioned to show the first occurrence of the dataname in the PROCEDURE DIVISION. DEF information in the DATA DIVISION is also highlighted.</p>
FX INDIRECT	<p>Allows you to see the indirect effect of a size change to a dataname one level at a time. If you use the SIZE operand without specifying the LEVELS operand (e.g., FX ZIP-CODE INDIRECT SIZE), a default value of ALL is assumed. All datanames or indirectly affected by a change in the size of the dataname are located. This results in a complete list of all datanames that need to be reviewed (for a size change of the specified dataname).</p> <p>If you specify the LEVELS operand with the SIZE operand (e.g., FX ZIP-CODE INDIRECT SIZE LEVELS 1), only the results of that many levels are highlighted. The ripple effect can be more clearly identified by starting with LEVELS 1, then increasing the number of levels to be searched.</p> <p>When using INDIRECT SIZE, any occurrence of an affected dataname in the USING clause of a CALL statement is tagged as a MOD. The called subprogram must be examined for other datanames that are affected as a result of the CALL. Use IN target to restrict the search to the specified IN target.</p> <p>When the REF operand is specified or used by default, all appropriate references are highlighted. If the reference is a DEFinition, USE, or MODification, the line is tagged accordingly. If the line contains a USE and a MODification, it is tagged as a REFerence.</p> <p>Use FX INDIRECT VALUE to find how a change in the logical use of a data item affects program.</p>
DBCS Strings	Supports DBCS strings.

Examples

To search backward for the previous occurrence of a use of DATE-CODE or its aliases, type this command:

```
FINDXTND DATE-CODE USE PREV
```

To highlight the definition of DATE-CODE in the DATA DIVISION, type this command:

```
FX DATE-CODE DEF NOALIAS ALL
```

To highlight all occurrences of the SWITCH field used in conditional statements, type this command:

```
FX SWITCH IN COND ALL
```

This example uses this code:

```
01  A.
05  B          PIC X.
05  C.
      10  D      PIC X.
      10  E      PIC X.
01  Z REDEFINES A.
05  Z1         PIC XXX.
66  L RENAMES Z.
```

If a FINDXTND command is entered for C with ALIAS specified:

- A is the parent.
- D and E are the children.
- Z is a redefine/rename.
- Z1 is a redefine/rename.
- L is a rename.

This example uses this code:

```
000010  READ INFILE INTO A.
000020  MOVE A TO B.
000030  MOVE B TO C.
000040  WRITE OUTFILE FROM C.
```

This FINDXTND command shows the possible origin of the data value and helps you determine where this data come from:

```
FX B MOD INDIRECT VALUE
```

The MOD operand is used to locate only those places where B is directly or indirectly set or modified. The fields in lines 10 and 20 are highlighted since B is modified by A on line 20, and A is modified by the READ statement on line 10.

This FINDXTND command shows all possible destinations of a data field and helps determine the places this value is used:

```
FX B USE INDIRECT VALUE
```

The USE operand is used to locate every place B is directly or indirectly used. The fields in lines 30 and 40 are highlighted because both statements use B. The above FINDXTND command shows that the value of B is used to modify C on line 30. On line 40, the value of C that was received from B is used.

This example uses this code:

```
000010 01 A.  
000020 05 B      PIC XX.  
.  
000070 01 C.  
000080 05 D      PIC XX.  
.  
001010      MOVE B TO X.  
001020      MOVE C TO A.  
001030      MOVE SPACES TO D.
```

This FINDXTND command is used to locate all groups whose definitions might have to be changed every time the definition for B changes:

```
FX B REF INDIRECT
```

The results are lines 10, 20, 70, 80, 1010, and 1020. Any change in the size of B is a change in the size of A as well. This means the definition for C should be changed since A now has a different record layout, and the statement on line 1020 would not execute as intended. D is also shown as a result because it is at the same offset within C as B is within A, and is the same size as B.

Note: _____

The number of results from FX INDIRECT REF depends greatly on the target name. For best results, select a target that is the smallest group item whose subfields are to be modified.

This example uses this code to show the LEVELS, VALUE, and SIZE operands.

```
000010  MOVE A TO B.
000020  MOVE X TO B.
000080  MOVE Y TO X.
000100  MOVE Y TO A.
000110  MOVE B TO C.
```

This command identifies the datanames in the program directly affected by a change in A. This list consists of only dataname A and its related aliases.

```
FX A INDIRECT LEVELS 0
```

This command identifies the statement (move A to B) that shows B might be affected by this move. This statement (line 10) and the definitions of A and B are highlighted.

```
FX A INDIRECT LEVELS 1
```

This command highlights the definitions of A, B, and X, plus the statements on lines 10 and 20.

```
FX A INDIRECT LEVELS 2
```

This command identifies all direct and indirect uses of the value of A. The statement on lines 100 and 110 would be highlighted.

```
FX A USE INDIRECT VALUE
```

This command identifies all direct and indirect modifications to the value of A. The statement on line 100 would be highlighted.

```
FX A MOD INDIRECT VALUE
```

This command identifies where A is affected by a change in the size of the target dataname. The statements on line 20 and 100 would be highlighted.

```
FX A USE INDIRECT SIZE
```

.label Function

The FINDXTND command supports the ISPF .label function with a target or an IN target.

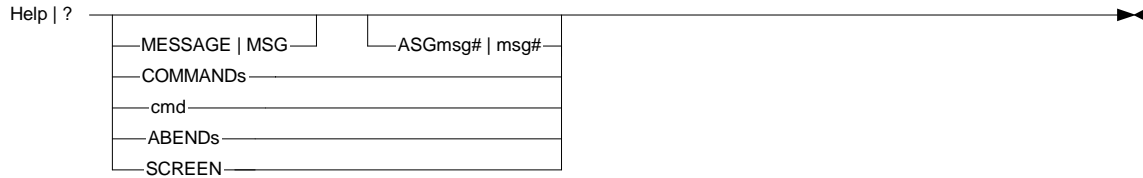
To use .label with a target, type this command:

```
FX .A
```

To use .label with an IN target (DATE-CODE), type this command:

```
FX DATE-CODE IN .A
```

HELP Command



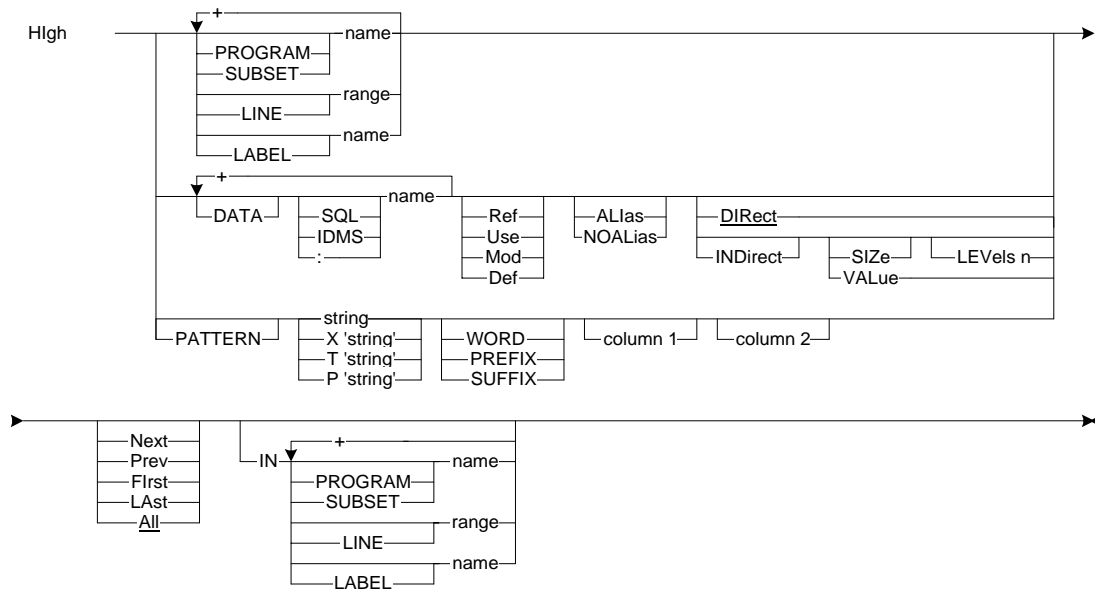
Function

Displays information about the current SmartEdit screen or pop-up, commands, messages, or abend codes.

Operands

Operand	Description
Blank	Displays the Help Tutorial for the current screen or pop-up, which describes all fields on the screen or pop-up and any special processing considerations. If a message displays, typing HELP with no operands displays the Help Explanation and Action screen for the current message.
MESSAGE MSG	Displays the Help Explanation and Action screen, which shows the specified short and long message, an explanation of the current message, and any actions to be performed.
ASGmsg# msg#	Specifies the ASG message number for which the Help Explanation and Action screen is to be displayed. This number consists of 1 to 4 digits. It is not necessary to enter leading zeros.
COMMANDs	Displays a list of all SmartEdit primary commands. Information for a particular command can then be displayed by selecting the appropriate number.
cmd	Specifies a SmartEdit primary command. When help is requested for a command, a long message displays giving a brief description of that command. Requesting help again then displays more detailed information about the command.
ABENDs	Displays the Abends screen that lists the types of abend codes.
SCREEN	Displays help for the current screen or pop-up, which describes all fields on the screen or pop-up and any special processing considerations.

HIGH Command



Function

Highlights source code lines containing the specified targets. Lines that are already highlighted are not reset.

Operands

Operand	Description																																										
PROGRAM <i>name</i>	Specifies the name of the main program or any nested program representing all the code contained in the program. This includes all the programs physically nested inside the specified program.																																										
SUBSET <i>name</i>	<p>Specifies a predefined COBOL language subset. These are the predefined COBOL language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CALL</td><td>DIRECTive</td><td>LABEL</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTion</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table> <p>A screen subset:</p> <table><tr><td>Highlighted</td><td> </td><td>HI</td></tr><tr><td>NONHighlighted</td><td> </td><td>NHI</td></tr><tr><td>Excluded</td><td> </td><td>X</td></tr><tr><td>NONExcluded</td><td> </td><td>NX</td></tr></table> <p>A tagged lines subset of tags displaying in columns 73 through 80.</p> <p>See "Product Overview" on page 5 for a description of each subset.</p>	ASsignment	DEFinition	IO	CALL	DIRECTive	LABEL	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTion	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS		Highlighted		HI	NONHighlighted		NHI	Excluded		X	NONExcluded		NX
ASsignment	DEFinition	IO																																									
CALL	DIRECTive	LABEL																																									
CIcs	DIVision	MATH																																									
COBOLII	DL/I	Output																																									
COBOL/370	DML	PARagraph																																									
COMment	ENtry	PERform																																									
CONditional	EXIt	SECTion																																									
DB2 SQL	GOto	SORTMerge																																									
DDL	IDMS	STructure																																									
DEBug	IMS																																										
Highlighted		HI																																									
NONHighlighted		NHI																																									
Excluded		X																																									
NONExcluded		NX																																									
LINE <i>range</i>	Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor screen.																																										
LABEL <i>name</i>	Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.																																										
DATA <i>name</i>	<p>Specifies a COBOL dataname or qualified COBOL dataname. Dataname refers to any valid COBOL reference for a data element.</p> <p>These subordinate operands apply to the dataname and are used to further qualify the HIGH command: REF, USE, MOD, DEF, ALIAS, NOALIAS, DIRECT, and INDIRECT. The defaults are REF, ALIAS, and DIRECT.</p>																																										
SQL <i>name</i>	Includes datanames that are DB2/SQL variables only.																																										
IDMS <i>name</i>	Includes datanames that are IDMS variables only.																																										

Operand	Description
: <i>name</i>	Includes datanames that are COBOL variables only.
Ref	Includes occurrences of the dataname where its value is defined, tested, used, set, or modified. This is the default value for the DATA name operand.
Use	Includes occurrences of the dataname where its value is being tested or used.
Mod	Includes occurrences of the dataname where its value is being set or modified.
Def	Includes definitions of the dataname in the DATA DIVISION.
ALias	Includes occurrences of the aliases for the dataname. These aliases include: <ul style="list-style-type: none"> • Parent - higher level group item • Child - lower level item • Rename/Redefinition - renamed, redefined, or 88 level items This is the default value for the DATA name operand.
NOAlias	Ignores aliases for the specified dataname.
DIRECT	Includes occurrences of the dataname (and aliases if specified) where it is directly tested, used, set, modified, or defined (based on the REF, USE, MOD, or DEF selection). The default for the DATA name operand is direct; however, if the SIZE, VALUE, or LEVELS operand is specified, INDIRECT is assumed.
INDirect	Includes any dataname indirectly affected by the specified dataname (and aliases if specified). You can further qualify the indirect data item using the SIZE, VALUE, and LEVELS subordinate operands. These subordinate operands indicate the type of indirect reference to be located. SIZE and ALL LEVELS are the defaults for the INDIRECT operand. S
SIZE	Includes datanames that could be directly or indirectly affected if the size of the specified dataname were to be changed. SIZE is only valid with the INDIRECT operand. This is the default for the INDIRECT operand.
VALue	Includes occurrences of datanames that are directly or indirectly affected by a change in the value of the specified dataname. The LEVELS subordinate operand is not used with VALUE. VALUE is only valid with the INDIRECT operand.

Operand	Description
LEVelS <i>n</i>	Includes all data items that are affected within the specified number of indirect levels. The VALUE subordinate operand is not used with LEVELS. ALL LEVELS is the default for the LEVELS operand. LEVELS is only valid with the INDIRECT operand.
PATTERN	Specifies an optional keyword indicating the characters that follow are part of a string.
<i>string</i>	Specifies a string of alphanumeric or DBCS characters. If the pattern string contains blanks, it must be enclosed in single or double quotes. You can further qualify the pattern string using the WORD, PREFIX, or SUFFIX subordinate operands. These subordinate operands describe how the pattern string is to be used.
X' <i>string</i> '	Specifies the hexadecimal string option. Specific unprintable characters may be specified by giving the EBCDIC hexadecimal value. The string must be enclosed in single or double quotes.
T' <i>string</i> '	Specifies the text string option. A character string may be entered regardless of upper or lowercase by using the text option. The string must be enclosed in single or double quotes.
P' <i>string</i> '	Specifies the picture string option. You can enter a string profile instead of exact characters. Enclose the string in single or double quotes. Nine special characters are defined by SmartEdit for use in picture strings. These special characters can be combined with other characters: <p>P'=' Any character</p> <p>P'¬' Any nonblank character</p> <p>P'.' Any nondisplay character</p> <p>P'#' Any numeric character</p> <p>P'-' Any non-numeric character</p> <p>P'@' Any alphabetic character (upper or lowercase)</p> <p>P'<' Any lowercase alphabetic character</p> <p>P'>' Any uppercase alphabetic character</p> <p>P'\$' Any special character (not alphabetic or numeric)</p>
WORD	Specifies the specified pattern string preceded and followed by any non-alphanumeric character (except a hyphen).
PREFIX	Specifies a word that begins with the specified pattern string.
SUFFIX	Specifies a word that ends with the specified pattern string.

Operand	Description
Column1	Specifies the column number where the search is to begin.
Column2	Specifies the column number where the search is to end.
Next	Searches forward from the current cursor position to the next occurrence of the requested target.
Prev	Searches backward from the current cursor position to the previous occurrence of the requested target.
First	Searches from the top of the source file to the first occurrence of the requested target.
LAst	Searches backward from the bottom of the source file to the first occurrence of the requested target.
All	Searches for all occurrences of the requested target and displays the number found in the short/long message field. This is the default value for the HIGH command.
IN	Restricts the HIGH command to the specified target type. This operand is optional.

Usage Notes

Source code lines can also be highlighted by selecting a Search pop-up on the Search pull-down. See ["Search" on page 107](#) for more information.

To concatenate targets, place a plus sign (+) between the target names, for example:

IO + CALL

These rules apply to concatenation:

- You can use a concatenated dataname wherever a dataname is valid. Dataname subordinate operands (REF, ALIAS, etc.) pertain to all datanames in a concatenated series.
- You can use a concatenated set name wherever a set name is valid. These sets are valid:

LABEL LINE PROGRAM SUBSET

For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times by using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, you can qualify it as P120-READ OF VIAFDEM1.

You can add lines to a highlighted set. For example, to see all PERFORM statements, then to add the conditional statements related to one of the PERFORM statements, use FINDXTND PERFORM, then use HIGH CONDITIONAL to add the additional lines.

To step through all IO statements, type HIGH IO NEXT and repeat the command until the end of the program is located.

Examples

To highlight all lines containing IO statements, type this command:

```
HIGH IO
```

Existing highlighted lines are not reset when you issue this command.

To highlight only labels within the highlighted lines, type this command:

```
FX LAB IN HI
```

All other highlighted lines are reset.

The HIGH command supports the ISPF .label function with a target or an IN target.

To use .label with a target, type this command:

```
HI .A
```

To use .label with an IN target (DATE-CODE), type this command:

```
HI DATE-CODE IN .A
```

JUMP Command

JUMP 

Function

Returns to the editor screen from the Tree View screen at the location corresponding to the cursor on the Tree View screen.

Operands

None.

Usage Notes

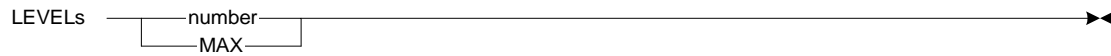
To return to the editor screen from the Tree View screen, follow this step:

- ▶ Select View ▶ Jump, or use the J line command.

The JUMP command is used in conjunction with the cursor location. Type JUMP in the command input area, and position the cursor to the line containing the target of the jump and press Enter.

See ["View" on page 87](#) for more information about the Tree View screen and the View pull-down.

LEVELS Command



Function

Redisplays the current Tree View screen to the specified level.

Operands

Operand	Description
Blank	Redisplays the current screen to the default level 1.
number	Redisplays the current screen to the level depth specified.
MAX	Redisplays the current screen to the maximum number of levels in the program.

Usage Notes

You can also change levels by selecting View ► Levels, which displays the View - Levels Request pop-up. See ["View" on page 87](#) for more information.

The LEVELS command redisplay the current screen to the level specified. If there are less than the number specified, they are expanded to the maximum level. If there are levels greater than the number specified, they are removed. The LEVELS command works globally throughout the displayed program.

LIST Command



Function

Displays the specified pop-up that lists pertinent information about the program being viewed. List pop-ups are available for each operand.

Operands

Operand	Description
Blank	Displays the List Menu pop-up that allows the user to select one of the other List pop-ups.
Calls	Displays the List - CALL Statements pop-up that lists all COBOL statements containing CALLs to other programs in the active program.
Equates	Displays the List - Equates pop-up that lists all equates for the active program.
Performs	Displays the List - Perform Range Names pop-up that lists all COBOL perform ranges in the active program.
PPrograms	Displays the List - Program/Subprogram Names pop-up that lists all subprograms and their calling hierarchy for COBOL II Release 3 or later programs containing subprograms.
Subsets	Displays the List - COBOL Subset Names pop-up that lists all subsets, along with a brief description of each.

Usage Notes

The LIST command is not available in Tree view.

List pop-ups are also available by selecting an action on the List pull-down. See ["List" on page 147](#) for more information about List pop-ups and the List pull-down.

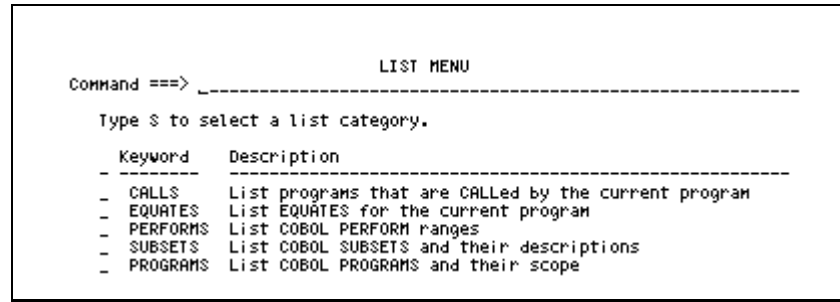
You can type the LPRINT * command (see ["LPRINT Command" on page 233](#) for more information) on any List pop-up to copy its contents to the List file.

List Menu Pop-up

To display the List Menu pop-up

- 1 Type LIST with no operands. The List Menu pop-up, shown in [Figure 132](#), displays.

Figure 132 • List Menu Pop-up

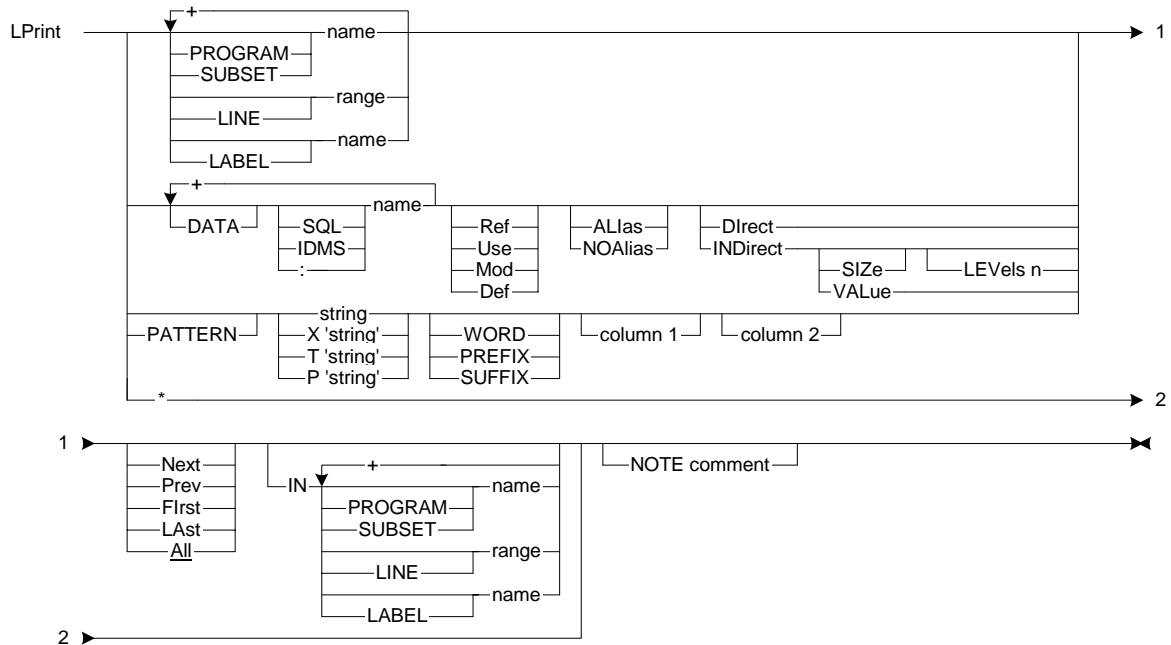


Keyword	Description
- CALLS	List programs that are CALLED by the current program
- EQUATES	List EQUATES for the current program
- PERFORMS	List COBOL PERFORM ranges
- SUBSETS	List COBOL SUBSETS and their descriptions
- PROGRAMS	List COBOL PROGRAMS and their scope

- 2 Select any item on this pop-up by typing S to the left of the desired keyword and pressing Enter. The pop-up for the selected option displays.

You can enter the LPRINT * command (see ["LPRINT Command" on page 233](#) for more information) on any List pop-up to copy its contents to the List file.

LPRINT Command



Function

Copies lines containing the requested target to the List file. The List file is processed on the Options - Log/List/Punch Definition pop-up (see [Figure 121 on page 165](#) for more information).

If the TSO PROFILE NOPREFIX option is in effect, the user ID is used as the high-level qualifier for the Log, List, and Punch files, for example:

```
userid.SEDyyyyy.VIAxxxxx,
```

where:

xxxxxx is log, list, or punch.

yyyyyy is the counter.

Operands

Operand	Description																																										
PROGRAM <i>name</i>	Specifies the name of the main program or any nested program representing all the code contained in the program. This includes all the programs physically nested inside the specified program.																																										
SUBSET <i>name</i>	<p>Specifies a predefined COBOL language subset. These are the predefined COBOL language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CALL</td><td>DIRECTive</td><td>LABel</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTion</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table> <p>A screen subset:</p> <table><tr><td>Highlighted</td><td> </td><td>HI</td></tr><tr><td>NONHighlighted</td><td> </td><td>NHI</td></tr><tr><td>Excluded</td><td> </td><td>X</td></tr><tr><td>NONExcluded</td><td> </td><td>NX</td></tr></table> <p>A tagged lines subset of tags displaying in columns 73 through 80.</p> <p>See "Subsets" on page 12 for a description of each subset.for a description of each subset.</p>	ASsignment	DEFinition	IO	CALL	DIRECTive	LABel	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTion	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS		Highlighted		HI	NONHighlighted		NHI	Excluded		X	NONExcluded		NX
ASsignment	DEFinition	IO																																									
CALL	DIRECTive	LABel																																									
CIcs	DIVision	MATH																																									
COBOLII	DL/I	Output																																									
COBOL/370	DML	PARagraph																																									
COMment	ENtry	PERform																																									
CONditional	EXIt	SECTion																																									
DB2 SQL	GOto	SORTMerge																																									
DDL	IDMS	STructure																																									
DEBug	IMS																																										
Highlighted		HI																																									
NONHighlighted		NHI																																									
Excluded		X																																									
NONExcluded		NX																																									
LINE <i>range</i>	Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor Screen.																																										
LABEL <i>name</i>	Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.																																										
DATA <i>name</i>	<p>Specifies a COBOL dataname or qualified COBOL dataname. Dataname refers to any valid COBOL reference for a data element.</p> <p>These subordinate operands apply to the dataname and can be used to further qualify the LPRINT command: REF, USE, MOD, DEF, ALIAS, NOALIAS, DIRECT, and INDIRECT. The defaults are REF, ALIAS, and DIRECT.</p>																																										
SQL <i>name</i>	Includes datanames that are DB2/SQL variables only.																																										
IDMS <i>name</i>	Includes datanames that are IDMS variables only.																																										

Operand	Description
: <i>name</i>	Includes datanames that are COBOL variables only.
Ref	Includes occurrences of the dataname where its value is defined, tested, used, set, or modified. This is the default value for the DATA name operand.
Use	Includes occurrences of the dataname where its value is being tested or used.
Mod	Includes occurrences of the dataname where its value is being set or modified.
Def	Includes definitions of the dataname in the DATA DIVISION.
ALias	Includes aliases for the dataname. These aliases include: <ul style="list-style-type: none"> • Parent - higher level group item • Child - lower level item • Rename/Redefinition - renamed, redefined, or 88 level items This is the default value for the DATA name operand.
NOAlias	Ignores aliases for the specified dataname.
DIRECT	Includes occurrences of the dataname (and aliases if specified) where it is directly tested, used, set, modified, or defined (based on the REF, USE, MOD, or DEF selection). The default for the DATA name operand is direct; however, if the SIZE, VALUE, or LEVELS operand is specified, INDIRECT is assumed.
INDirect	Includes any dataname indirectly affected by the specified dataname (and aliases if specified). You can further qualify the indirect data item using the SIZE, VALUE, and LEVELS subordinate operands. These subordinate operands indicate the type of indirect reference to be located. SIZE and ALL LEVELS are the defaults for the INDIRECT operand.
SIZE	Includes datanames that could be directly or indirectly affected if the size of the specified dataname were to be changed. SIZE is only valid with the INDIRECT operand. This is the default for the INDIRECT operand.
VALue	Includes occurrences of datanames that are directly or indirectly affected by a change in the value of the specified dataname. The LEVELS subordinate operand is not used with VALUE. VALUE is only valid with the INDIRECT operand.

Operand	Description
LEVelS <i>n</i>	Includes all data items that are affected within the specified number of indirect levels. The VALUE subordinate operand is not used with LEVELS. ALL LEVELS is the default for the LEVELS operand. LEVELS is only valid with the INDIRECT operand.
PATTERN	Specifies an optional keyword indicating the characters that follow are part of a string.
<i>string</i>	Specifies a string of alphanumeric or DBCS characters. If the pattern string contains blanks, it must be enclosed in single or double quotes. You can further qualify the pattern string using the WORD, PREFIX, or SUFFIX subordinate operands. These subordinate operands describe how the pattern string is to be used.
X' <i>string</i> '	Specifies the hexadecimal string option. Specific unprintable characters may be specified by giving the EBCDIC hexadecimal value. The string must be enclosed in single or double quotes.
T' <i>string</i> '	Specifies the text string option. A character string may be entered regardless of upper or lowercase by using the text option. The string must be enclosed in single or double quotes.
P' <i>string</i> '	Specifies the picture string option. A string profile may be entered instead of exact characters. The string must be enclosed in single or double quotes. Nine special characters are defined by SmartEdit for use in picture strings. Use these special characters to combine with other characters: <p>P'=' Any character</p> <p>P'¬' Any nonblank character</p> <p>P'.' Any nondisplay character</p> <p>P'#' Any numeric character</p> <p>P'-' Any non-numeric character</p> <p>P'@' Any alphabetic character (upper or lowercase)</p> <p>P'<' Any lowercase alphabetic character</p> <p>P'>' Any uppercase alphabetic character</p> <p>P'\$' Any special character (not alphabetic or numeric)</p>
WORD	Specifies the pattern string preceded and followed by any non-alphanumeric character (except a hyphen).
PREFIX	Specifies a word that begins with the specified pattern string.
SUFFIX	Specifies a word that ends with the specified pattern string.

Operand	Description
Column1	Specifies the column number where the search is to begin.
Column2	Specifies the column number where the search is to end.
* (asterisk)	Causes the entire virtual screen (all data that can be viewed by scrolling up and down) to be copied to the List file. All excluded lines are copied to the List file as excluded lines (as they display on the screen at the time you enter the LPRINT * command).
Next	Searches forward from the current cursor position to the next occurrence of the requested target.
Prev	Searches backward from the current cursor position to the previous occurrence of the requested target.
First	Searches from the top of the source file to the first occurrence of the requested target.
LAst	Searches backward from the bottom of the source file to the first occurrence of the requested target.
All	Searches for all occurrences of the requested target and displays the number found in the short/long message field. This is the default value for the LPRINT command.
IN	Restricts the LPRINT command to the specified target type. This operand is optional.
NOTE comment	Places descriptive comment lines in the List file. These comments are included in the printed output from the List file. Text of the comment can be a maximum of 50 alphanumeric or 23 DBCS characters.

Usage Notes

Lines can also be copied to the List file by selecting a Search pop-up on the Search pull-down. See ["Search" on page 107](#) for more information.

To concatenate targets, place a plus sign (+) between the target names, as shown in this example:

```
IO + CALL
```

These rules apply to concatenation:

- You can use a concatenated dataname wherever a dataname is valid. Dataname subordinate operands (REF, ALIAS, etc.) pertain to all datanames in a concatenated series.
- You can use a concatenated set name wherever a set name is valid. These sets are valid:

LABEL LINE PROGRAM SUBSET

For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times by using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, then it can be qualified as P120-READ OF VIAFDEM1.

LPRINT copies the specified target to the List file for subsequent printing.

You can use LPRINT * on these pop-ups to copy their contents to the List file:

- List - CALL Statements
- List - Equates
- List - Perform Range Names
- List - Program/Subprogram Names
- List - COBOL Subset Names

See ["Options" on page 155](#) for more information about the List file and the Options - Log/List/Punch Definition pop-up ([Figure 121 on page 165](#)). See ["View" on page 87](#) for more information about the View screens. See ["List" on page 147](#) for more information about the List pop-ups.

Examples

To copy all lines containing the characters ZIP to the List file for subsequent printing, type this command:

```
LPRINT ZIP
```

To copy the entire source file to the List file, type this command:

```
LPRINT *
```

The LPRINT command supports the ISPF .label function with a target range or an IN target range.

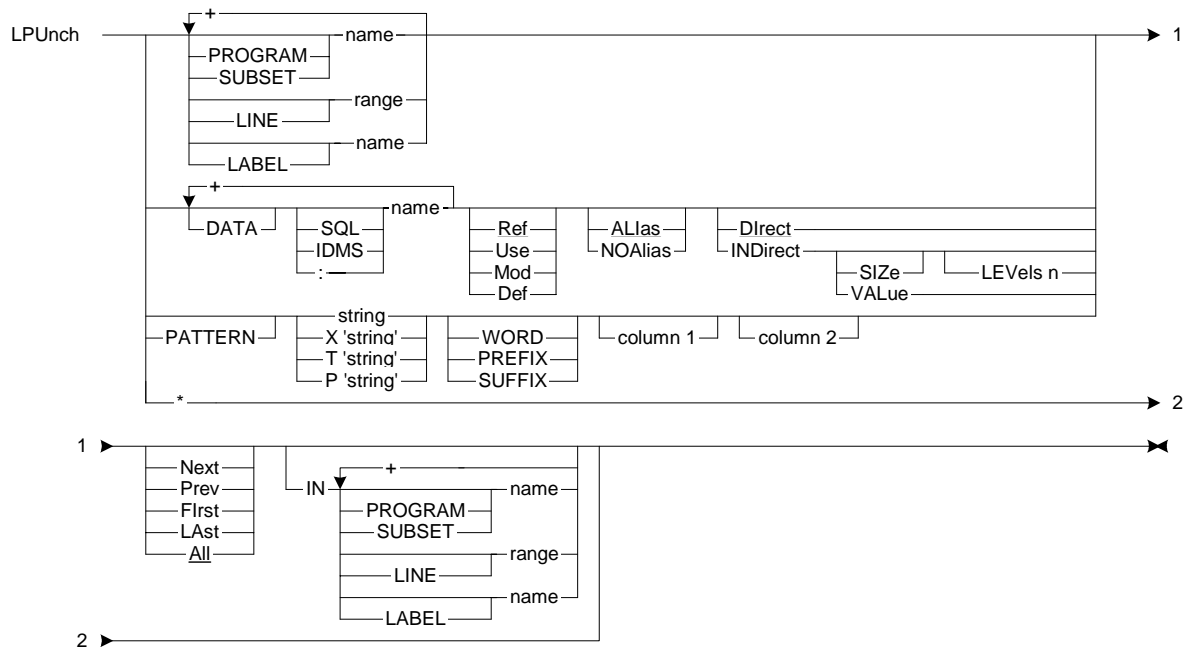
To copy lines within a range to the LIST file, type this command:

```
LP .A .B
```

To copy all input/output lines within a range to the LIST file, type this command:

```
LP IO IN .A .B
```

LPUNCH Command



Function

The LPUNCH command copies lines containing the specified target to the Punch file for subsequent processing. The Punch file is processed on the Options - Log/List/Punch Definition pop-up (see [Figure 121 on page 165](#) for more information).

If the TSO PROFILE NOPREFIX option is in effect, the user ID is used as the high-level qualifier for the Log, List, and Punch files. For example:

```
userid.SEDyyyyy.VIAxxxxxx
```

where:

xxxxxx is log, list, or punch.

yyyyyy is the counter.

Operands

Operand	Description																																										
PROGRAM <i>name</i>	Specifies the name of the main program or any nested program representing all the code contained in the program. This includes all the programs physically nested inside the specified program.																																										
SUBSET <i>name</i>	<p>Specifies a predefined COBOL language subset. These are the predefined COBOL language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CALL</td><td>DIRECTive</td><td>LABel</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTion</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table> <p>A screen subset:</p> <table><tr><td>Highlighted</td><td> </td><td>HI</td></tr><tr><td>NONHighlighted</td><td> </td><td>NHI</td></tr><tr><td>Excluded</td><td> </td><td>X</td></tr><tr><td>NONExcluded</td><td> </td><td>NX</td></tr></table> <p>A tagged lines subset of tags displaying in columns 73 through 80.</p> <p>See "Subsets" on page 12 for a description of each subset.</p>	ASsignment	DEFinition	IO	CALL	DIRECTive	LABel	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTion	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS		Highlighted		HI	NONHighlighted		NHI	Excluded		X	NONExcluded		NX
ASsignment	DEFinition	IO																																									
CALL	DIRECTive	LABel																																									
CIcs	DIVision	MATH																																									
COBOLII	DL/I	Output																																									
COBOL/370	DML	PARagraph																																									
COMment	ENtry	PERform																																									
CONditional	EXIt	SECTion																																									
DB2 SQL	GOto	SORTMerge																																									
DDL	IDMS	STructure																																									
DEBug	IMS																																										
Highlighted		HI																																									
NONHighlighted		NHI																																									
Excluded		X																																									
NONExcluded		NX																																									
LINE <i>range</i>	Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor screen.																																										
LABEL <i>name</i>	Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.																																										
DATA <i>name</i>	<p>Specifies a COBOL dataname or qualified COBOL dataname. Dataname refers to any valid COBOL reference for a data element.</p> <p>These subordinate operands apply to the dataname and can be used to further qualify the LPUNCH command: REF, USE, MOD, DEF, ALIAS, NOALIAS, DIRECT, and INDIRECT. The defaults are REF, ALIAS, and DIRECT.</p>																																										
SQL <i>name</i>	Includes datanames that are DB2/SQL variables only.																																										
IDMS <i>name</i>	Includes datanames that are IDMS variables only.																																										

Operand	Description
: <i>name</i>	Includes datanames that are COBOL variables only.
Ref	Includes occurrences of the dataname where its value is defined, tested, used, set, or modified. This is the default value for the DATA name operand.
Use	Includes occurrences of the dataname where its value is being tested or used.
Mod	Includes occurrences of the dataname where its value is being set or modified.
Def	Includes definitions of the dataname in the DATA DIVISION.
ALias	Includes occurrences of the aliases for the dataname. These aliases include: <ul style="list-style-type: none"> • Parent - higher level group item • Child - lower level item • Rename/Redefinition - renamed, redefined, or 88 level items This is the default value for the DATA name operand.
NOAlias	Ignores aliases for the specified dataname.
DIREct	Includes occurrences of the dataname (and aliases if specified) where it is directly tested, used, set, modified, or defined (based on the REF, USE, MOD, or DEF selection). The default for the DATA name operand is direct; however, if the SIZE, VALUE, or LEVELS operand is specified, INDIRECT is assumed.
INDirect	Includes any dataname indirectly affected by the specified dataname (and aliases if specified). You can further qualify the indirect data item using the SIZE, VALUE, and LEVELS subordinate operands. These subordinate operands indicate the type of indirect reference to be located. SIZE and ALL LEVELS are the defaults for the INDIRECT operand.
SIZE	Includes datanames that could be directly or indirectly affected if the size of the specified dataname were to be changed. SIZE is only valid with the INDIRECT operand. This is the default for the INDIRECT operand.
VALue	Includes occurrences of datanames that are directly or indirectly affected by a change in the value of the specified dataname. The LEVELS subordinate operand is not used with VALUE. VALUE is only valid with the INDIRECT operand.

Operand	Description
LEVelS <i>n</i>	Includes all data items that are affected within the specified number of indirect levels. The VALUE subordinate operand is not used with LEVELS. ALL LEVELS is the default for the LEVELS operand. LEVELS is only valid with the INDIRECT operand.
PATTERN	Specifies an optional keyword indicating the characters that follow are part of a string.
<i>string</i>	Specifies a string of alphanumeric or DBCS characters. If the pattern string contains blanks, it must be enclosed in single or double quotes. You can further qualify the pattern string using the WORD, PREFIX, or SUFFIX subordinate operands. These subordinate operands describe how the pattern string is to be used.
X' <i>string</i> '	Specifies the hexadecimal string option. Specific unprintable characters may be specified by giving the EBCDIC hexadecimal value. The string must be enclosed in single or double quotes.
T' <i>string</i> '	Specifies the text string option. A character string may be entered regardless of upper or lowercase by using the text option. The string must be enclosed in single or double quotes.
P' <i>string</i> '	Specifies the picture string option. A string profile may be entered instead of exact characters. The string must be enclosed in single or double quotes. Nine special characters are defined by SmartEdit for use in picture strings. These special characters combined with other characters: <p>P'=' Any character</p> <p>P'¬' Any nonblank character</p> <p>P'.' Any nondisplay character</p> <p>P'#' Any numeric character</p> <p>P'-' Any non-numeric character</p> <p>P'@' Any alphabetic character (upper or lowercase)</p> <p>P'<' Any lowercase alphabetic character</p> <p>P'>' Any uppercase alphabetic character</p> <p>P'\$' Any special character (not alphabetic or numeric)</p>
WORD	Specifies the pattern string preceded and followed by any non-alphanumeric character (except a hyphen).
PREFIX	Specifies a word that begins with the specified pattern string.
SUFFIX	Specifies a word that ends with the specified pattern string.

Operand	Description
Column1	Specifies the column number where the search is to begin.
Column2	Specifies the column number where the search is to end.
* (asterisk)	Causes the entire virtual screen (all data that can be viewed by scrolling up and down) to be copied to the Punch file. All excluded lines are copied to the Punch file as excluded lines (as they display on the screen at the time the LPUNCH * command is entered).
Next	Searches forward from the current cursor position to the next occurrence of the requested target.
Prev	Searches backward from the current cursor position to the previous occurrence of the requested target.
Ffirst	Searches from the top of the source file to the first occurrence of the requested target.
LAst	Searches backward from the bottom of the source file to the first occurrence of the requested target.
All	Searches for all occurrences of the requested target and displays the number found in the short message field. This is the default value for the LPUNCH command.
IN	Restricts the LPUNCH command to the specified target type. This operand is optional.

Usage Notes

Lines can also be copied to the Punch file by selecting a Search pop-up on the Search pull-down. See ["Search" on page 107](#) for more information.

LPUNCH copies the specified targets to the Punch file for subsequent processing.

To concatenate targets, place a plus sign (+) between the target names, as shown in this example:

```
IO + CALL
```

These rules apply to concatenation:

- You can use a concatenated dataname wherever a dataname is valid. Dataname subordinate operands (REF, ALIAS, etc.) pertain to all datanames in a concatenated series.
- You can use a concatenated set name wherever a set name is valid. These sets are valid:

LABEL LINE PROGRAM SUBSET

For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times by using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, you could qualify it as P120-READ OF VIAFDEM1.

See ["Options" on page 155](#) for more information about the Punch file and the Options - Log/List/Punch Definition pop-up ([Figure 121 on page 165](#)).

Examples

To copy all input and output statements in the current program to the Punch file, type this command:

LPUNCH IO

The LPUNCH command supports the ISPF .label function with a target or an IN target.

To copy lines within a range to the Punch file, type this command:

LPU .A .B

To copy all input/output lines within a range to the PUNCH file, type this command:

LPU IO IN .A .B

PARMDEF Command

PARMDEF | PDEF

Function

The PARMDEF command displays the Options - Product Parameters pop-up (see [Figure 114 on page 157](#) for more information). The Options - Product Parameters pop-up is used to set parameters that affect the online operation of SmartEdit.

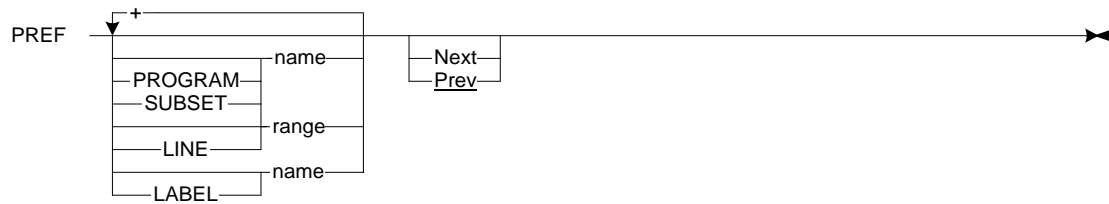
Operands

None.

Usage Notes

You can also display the Options - Product Parameters pop-up by selecting Options ► Product Parameters (See ["Options Pull-down" on page 156](#)).

PREF Command



Function

Locates CALLED subprograms that are separate modules residing in the same source library as the main program. This command displays the View - Paragraph Cross Reference pop-up for the requested target. The View - Paragraph Cross Reference pop-up shows how control is transferred to or from the target paragraphs.

Operands

Operand	Description
Blank	Specifies that the target paragraph set is determined by the cursor location when the PREF command is issued.
PROGRAM <i>name</i>	Specifies any internal subprogram in COBOL Release 3 (or later) programs.

Operand	Description																																						
SUBSET <i>name</i>	<p>Specifies a predefined COBOL language subset. These are the predefined COBOL language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CAIl</td><td>DIREctive</td><td>LABel</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTion</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table> <p>A screen subset:</p> <table><tr><td>Highlighted</td><td> HI</td></tr><tr><td>NONHighlighted</td><td> NHI</td></tr><tr><td>Excluded</td><td> X</td></tr><tr><td>NONExcluded</td><td> NX</td></tr></table> <p>A tagged lines subset of tags displaying in columns 73 through 80.</p>	ASsignment	DEFinition	IO	CAIl	DIREctive	LABel	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTion	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS		Highlighted	HI	NONHighlighted	NHI	Excluded	X	NONExcluded	NX
ASsignment	DEFinition	IO																																					
CAIl	DIREctive	LABel																																					
CIcs	DIVision	MATH																																					
COBOLII	DL/I	Output																																					
COBOL/370	DML	PARagraph																																					
COMment	ENtry	PERform																																					
CONditional	EXIt	SECTion																																					
DB2 SQL	GOto	SORTMerge																																					
DDL	IDMS	STructure																																					
DEBug	IMS																																						
Highlighted	HI																																						
NONHighlighted	NHI																																						
Excluded	X																																						
NONExcluded	NX																																						
LINE <i>range</i>	<p>Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor screen.</p>																																						
LABEL <i>name</i>	<p>Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.</p>																																						
Next	<p>Generates a list of paragraphs to which the target paragraph transfers control.</p>																																						
Prev	<p>Generates a list of paragraphs that transfer control to the target paragraph. This is the default for the PREF command.</p>																																						

Usage Notes

You can also display the View - Paragraph Cross Reference pop-up by selecting View ► Paragraph X-Ref.

The PREF command includes all paragraphs containing the target lines. For example, PREF IO displays every paragraph containing IO statements.

To concatenate targets, place a plus sign (+) between the target names, as shown in this example:

```
IO + CALL
```

You can use a concatenated set name wherever a set name is valid. These sets are valid:

```
LABEL    LINE    PROGRAM    SUBSET
```

For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times by using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, you could qualify it as P120-READ OF VIAFDEM1.

When there is a blank target with the cursor located on excluded lines, these items apply:

- The first line of the excluded block is implicitly selected.
- If the selected line is not a label, then a backward search is done to locate the previous label. This label then becomes the target.

See ["View" on page 87](#) for more information about the View - Paragraph Cross Reference pop-up.

Example

This command displays the View - Paragraph Cross Reference pop-up with all paragraphs that transfer control to the WRITE-ROUTINE paragraph:

```
PREF WRITE-ROUTINE
```

The PREF command supports the ISPF .label function with a target range.

To perform a paragraph cross-reference within a range, type this command:

```
PRE .A .B
```

PREPROC Command

PREPROC 

Function

The PREPROC command displays the Options - Preprocessors pop-up (see [Figure 124 on page 172](#)). The Options-Preprocessors pop-up is used to select the preprocessor group to be used in your SmartEdit session.

Operands

None.

Usage Notes

You can also display the Options - Preprocessors pop-up by selecting Options ► Preprocessors.

PRINTLOG Command

PRINTLOG | PLOg 

Function

Displays the Options - Log/List/Punch Definition pop-up (see [Figure 121 on page 165](#) for more information).

Operands

None.

Usage Notes

You can also display the Options - Log/List/Punch Definition pop-up by selecting Options ► Log/List/Punch (see ["Options Pull-down" on page 156](#)).

Issue the PRINTLOG command from any SmartEdit screen. The Log, List, and Punch files are processed on the Options - Log/List/Punch Definition pop-up.

PRINTLST Command

PRINTLst | PList



Function

Displays the Options - Log/List/Punch Definition pop-up (see [Figure 121 on page 165](#) for more information).

Operands

None.

Usage Notes

You can also display the Options - Log/List/Punch Definition pop-up by selecting Options ► Log/List/Punch (see ["Options Pull-down" on page 156](#) for more information).

You can issue the PRINTLST command from any SmartEdit screen. The Log, List, and Punch files are processed on the Options - Log/List/Punch Definition pop-up.

PRODLVL Command

PRODLVL _____ ▶◀

Function

Displays the current SmartEdit and Center product level.

Operands

None.

Usage Notes

This information is requested when you contact ASG Customer Support for assistance. Selecting Help ► About displays the Help - About pop-up. See ["Help - About Pop-up" on page 179](#) for more information.

The PRODLVL command displays the product name, operating system, product release number, and release level on the message line, in this format:

```
ASG1554I ASG-SmartEdit-OS(XA) Rn.n AT Lnnn, ASG-CENTER Rn.n AT Lnnn
```

where:

Rn . n is the release number.

Lnnn is the release level.

RECALL Command



Function

Displays the previous primary command, message, or pop-up. The last 20 commands that have been executed and the last 20 messages that have been displayed are stacked. These commands or messages redisplay using the RECALL command.

Operands

Operand	Description
Blank	Displays the last primary command that was stacked. After the RECALL command has been entered with operands, typing RECALL with no operands reuses the same operands that were last entered.
COMmand CMD	Displays a stacked primary command. This is the default.
MESsage MSG	Displays a stacked message.
NEXT	Displays the next command or message in the stack.
PREV	Displays the previous command or message in the stack. This is the default value.
POPup	Displays the pop-up that was most recently requested from a pull-down.

Usage Notes

You can enter the RECALL command repeatedly to display any of the 20 stacked commands or messages. Use the NEXT and PREV operands to move forward or backward through the stacked commands or messages. After the desired command displays, execute it again by pressing Enter. You can change any command that is recalled prior to executing it. To re-execute the last saved primary command again without modification, you can use the REPEAT command.

The operands specified for the RECALL command remain in effect until one of these conditions occur:

- A different operand is specified.
- A different primary command is executed. When this occurs, the RECALL command default operands are automatically set. A message displays that indicates all stacked commands or messages have been shown and the stack is being redisplayed.

These commands are not stacked for use by the RECALL command:

- COBEDIT
- HELP
- PRODLVL
- RECALL
- RHIGH
- REPEAT
- RSCROLL

FIND Example

This example assumes these seven commands were entered for the current program and that seven messages were displayed:

- X ALL
- ZOOMIN
- PREF
- ZOOMOUT
- FX
- SCROLL
- FX

This command displays the previous message (number 7):

```
RECALL MSG
```

The MSG operand remains in effect until the COMMAND or CMD operand is specified. Therefore, typing RECALL without changing the operands displays the previous message (number 6).

Type this command to display the previous command (number 7):

```
RECALL CMD
```

The CMD operand remains in effect until the MSG operand is specified. Therefore, typing RECALL without changing the operands displays the previous command (number 6).

Type this command to display the next command (number 7):

```
RECALL CMD NEXT
```

REDO Command

REDO 

Function

Executes the corresponding repeat command from the cursor position, as shown in this table:

Last Command	Command Executed
FINDXTEND	RFIND
HIGH	RHIGH
PREF	RPREF
SCROLL	RSCROLL
TREEVIEW	RTREEVW

Only these commands are re-executed when you type REDO.

Operands

None.

Usage Notes

REDO automatically executes the corresponding repeat command, depending on the last command entered. For example, if the last command entered was FINDXTND, RFIND is executed.

REFRESH Command

REFresh 

Function

Provides fresh versions of copy members.

Operands

None.

Usage Notes

Refresh Copy members by selecting Options ► Refresh. See ["Options" on page 155](#) for more information.

The REFRESH command redisplay the Options - COPY/Include Libraries screen ([Figure 11 on page 24](#)) so the Copy Library information can be updated, then reads the most recent COPY members. COPY members are retrieved from partitioned datasets and from Librarian and Panvalet datasets. The ++INCLUDE lines are retrieved from Panvalet datasets only. The -INC lines are retrieved from Librarian datasets only.

Note:

When editing a Panvalet or Librarian member, the ++INCLUDE or -INC expansion is set to NO. This allows SmartEdit to do the expansion when necessary.

REPEAT Command

REPEat 

Function

Executes the last ESW stacked primary command again.

Operands

None.

Usage Notes

Type REPEAT in the command input area to re-execute the last stacked ESW primary command. The command is executed from the cursor position.

To modify the command before it is re-executed, use the RECALL command.

Example

Type this command to display the next occurrence of a highlighted line:

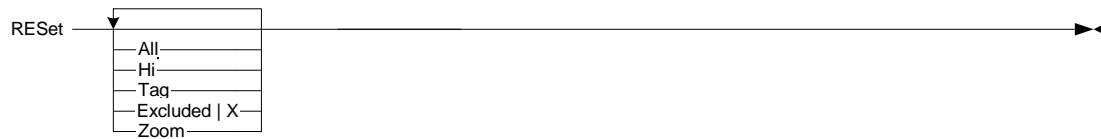
```
SCROLL HI NEXT
```

Instead of re-entering the same command, you can type this command:

```
REPEAT
```

The command above executes the SCROLL HI NEXT command again from the cursor position.

RESET Command



Function

Removes highlighting, erases tags in columns 1 through 6 or 73 through 80, and redisplay excluded lines. You can specify any combination of operands.

The editor accepts all ISPF RESET command operands in addition to the SmartEdit RESET command operands.

Operands

Operand	Description
Blank	Defaults to the value ALL.
All	Resets all conditions indicated by each of the other operands. This is the default.
Hi	Removes the highlighting from any line that has been highlighted as a result of another SmartEdit command.
Tag	Erases tags in columns 1 through 6 that have been set as a result of another SmartEdit command.
Excluded X	Displays any lines that have been excluded.
Zoom	Collapses any copy members that have been expanded.

Usage Notes

You can also reset features by selecting View ► Reset, which displays the View - Reset Request pop-up (see [Figure 80 on page 103](#) for more information).

The SmartEdit RESET command also accepts all operands that are valid in the ISPF RESET command. If the ISPF/PDF RESET ALL editor command is entered, highlighting is reset along with all other data.

RFIND Command

RFIND 

Function

The RFIND command repeats the last FINDXTND or ISPF FIND command from the cursor position.

Operands

None.

Usage Notes

The RFIND command is a convenient means of locating the next occurrence of the specified target of a FINDXTND or ISPF FIND command. The search is performed in the direction indicated in the FINDXTND or FIND command.

The REDO command can also be used to execute the RFIND command. See ["REDO Command" on page 257](#) for more information.

RHIGH Command

RHlgh 

Function

Repeats the last HIGH command from the cursor position.

Operands

None.

Usage Notes

Type RHIGH in the command input area to repeat the last HIGH command from the cursor position.

You can also use the REDO command to execute the RHIGH command. See ["REDO Command" on page 257](#) for more information.

RPREF Command

RPref 

Function

From the editor screen, the RPREF command redisplay the last View - Paragraph Cross Reference pop-up. From the View - Paragraph Cross Reference pop-up, the RPREF command redisplay the last editor screen.

Operands

None.

Usage Notes

The View - Paragraph Cross Reference pop-up is redisplayed with the exact information as previously shown. Use RPREF to go back and forth between the editor screen and the View - Paragraph Cross Reference pop-up.

When you enter RPREF without entering a PREF command, RPREF functions as a PREF command with the current cursor location as the target and default direction.

You can also use the REDO command to execute the RPREF command. See ["REDO Command" on page 257](#) for more information.

RSCROLL Command

RSCroll 

Function

Repeats the last SCROLL command from the cursor position.

Operands

None.

Usage Notes

Type RSCROLL in the command input area to repeat the last SCROLL command from the cursor position.

The REDO command can also be used to execute the RSCROLL command. See ["REDO Command" on page 257](#) for more information.

RTREEVW Command

RTREEvw | RTV 

Function

Returns to the Tree View screen after displaying the editor screen from Tree View.

Operands

None.

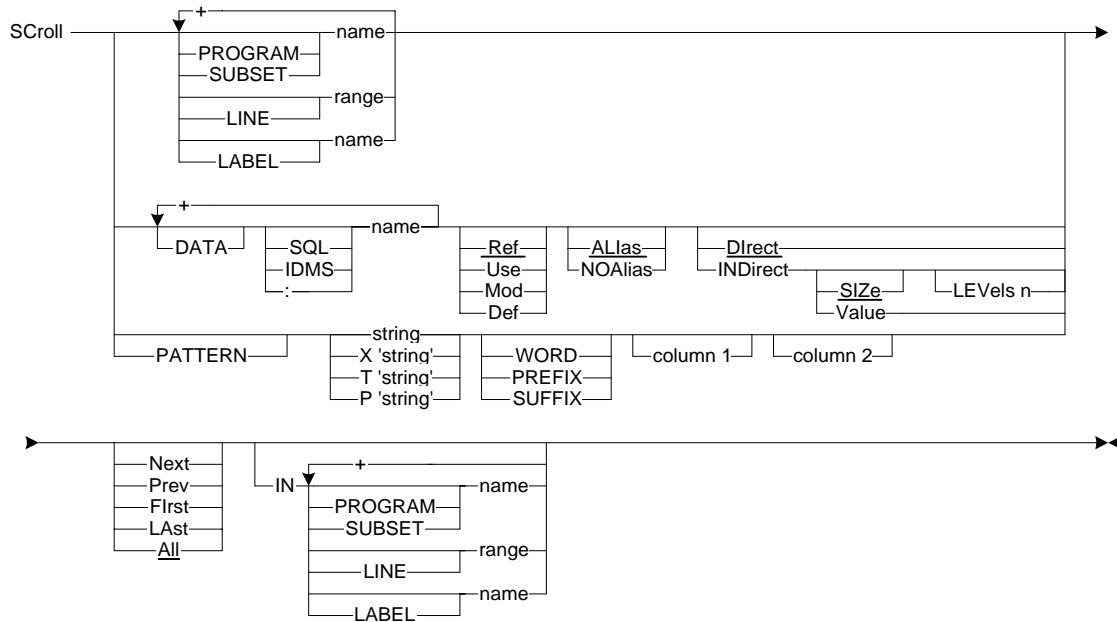
Usage Notes

The Tree View screen is redisplayed with the exact information as previously shown.

The REDO command can also be used to execute the RTREEVW command. See ["REDO Command" on page 257](#) for more information.

If source is changed or another edit session was executed, RTREEVW is invalid. A new TREEVIEW command must be issued.

SCROLL Command



Function

Performs an intelligent search of the source code for one or all occurrences of a specified target and positions the screen to the first line containing the specified target. Highlighted lines remain unchanged.

Operands

Operand	Description																																										
PROGRAM <i>name</i>	Specifies the name of the main program or any nested program representing all the code contained in the program. This includes all the programs physically nested inside the specified program.																																										
SUBSET <i>name</i>	<p>Specifies a predefined COBOL language subset. These are the predefined COBOL Language subset names:</p> <table><tr><td>ASsignment</td><td>DEFinition</td><td>IO</td></tr><tr><td>CAll</td><td>DIRECTive</td><td>LABel</td></tr><tr><td>CIcs</td><td>DIVision</td><td>MATH</td></tr><tr><td>COBOLII</td><td>DL/I</td><td>Output</td></tr><tr><td>COBOL/370</td><td>DML</td><td>PARagraph</td></tr><tr><td>COMment</td><td>ENtry</td><td>PERform</td></tr><tr><td>CONditional</td><td>EXIt</td><td>SECTion</td></tr><tr><td>DB2 SQL</td><td>GOto</td><td>SORTMerge</td></tr><tr><td>DDL</td><td>IDMS</td><td>STructure</td></tr><tr><td>DEBug</td><td>IMS</td><td></td></tr></table> <p>A screen subset:</p> <table><tr><td>Highlighted</td><td> </td><td>HI</td></tr><tr><td>NONHighlighted</td><td> </td><td>NHI</td></tr><tr><td>Excluded</td><td> </td><td>X</td></tr><tr><td>NONExcluded</td><td> </td><td>NX</td></tr></table> <p>A tagged lines subset of tags displaying in columns 73 through 80.</p> <p>See "Product Overview" on page 5 for a description of each subset.</p>	ASsignment	DEFinition	IO	CAll	DIRECTive	LABel	CIcs	DIVision	MATH	COBOLII	DL/I	Output	COBOL/370	DML	PARagraph	COMment	ENtry	PERform	CONditional	EXIt	SECTion	DB2 SQL	GOto	SORTMerge	DDL	IDMS	STructure	DEBug	IMS		Highlighted		HI	NONHighlighted		NHI	Excluded		X	NONExcluded		NX
ASsignment	DEFinition	IO																																									
CAll	DIRECTive	LABel																																									
CIcs	DIVision	MATH																																									
COBOLII	DL/I	Output																																									
COBOL/370	DML	PARagraph																																									
COMment	ENtry	PERform																																									
CONditional	EXIt	SECTion																																									
DB2 SQL	GOto	SORTMerge																																									
DDL	IDMS	STructure																																									
DEBug	IMS																																										
Highlighted		HI																																									
NONHighlighted		NHI																																									
Excluded		X																																									
NONExcluded		NX																																									
LINE <i>range</i>	Specifies a single line number or range of lines. Line numbers are displayed in columns 1 through 6 of the editor screen. When a range is specified, the data is scrolled to the first line in the range.																																										
LABEL <i>name</i>	Specifies any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE and PROC literals can also be specified.																																										
DATA <i>name</i>	<p>Specifies a COBOL dataname or qualified COBOL dataname. Dataname refers to any valid COBOL reference for a data element.</p> <p>These subordinate operands apply to the dataname and can be used to further qualify the SCROLL command: REF, USE, MOD, DEF, ALIAS, NOALIAS, DIRECT, and INDIRECT. The defaults are REF, ALIAS, and DIRECT.</p>																																										
SQL <i>name</i>	Includes datanames that are DB2/SQL variables only.																																										

Operand	Description
IDMS <i>name</i>	Includes datanames that are IDMS variables only.
: <i>name</i>	Includes datanames that are COBOL variables only.
Ref	Includes occurrences of the dataname where its value is defined, tested, used, set, or modified. This is the default value for the DATA name operand.
Use	Includes occurrences of the dataname where its value is being tested or used.
Mod	Includes occurrences of the dataname where its value is being set or modified.
Def	Includes definitions of the dataname in the DATA DIVISION.
ALias	Includes aliases for the dataname. These aliases include: <ul style="list-style-type: none"> • Parent - higher level group item • Child - lower level item • Rename/Redefinition - renamed, redefined, or 88 level items This is the default value for the DATA name operand.
NOAlias	Ignores aliases for the specified dataname.
DIRECT	Includes occurrences of the dataname (and aliases if specified) where it is directly tested, used, set, modified, or defined (based on the REF, USE, MOD, or DEF selection). The default for the DATA name operand is direct; however, if the SIZE, VALUE, or LEVELS operand is specified, INDIRECT is assumed.
INDirect	Includes any dataname indirectly affected by the specified dataname (and aliases if specified). You can further qualify the indirect data item using the SIZE, VALUE, and LEVELS subordinate operands. These subordinate operands indicate the type of indirect reference to be located. SIZE and ALL LEVELS are the defaults for the INDIRECT operand.
SIZE	Includes datanames that could be directly or indirectly affected if the size of the specified dataname were to be changed. SIZE is only valid with the INDIRECT operand. This is the default for the INDIRECT operand.

Operand	Description
VALue	Includes occurrences of datanames that are directly or indirectly affected by a change in the value of the specified dataname. The LEVELS subordinate operand is not used with VALUE. VALUE is only valid with the INDIRECT operand.
LEVel s n	Includes all data items that are affected within the specified number of indirect levels. The VALUE subordinate operand is not used with LEVELS. ALL LEVELS is the default for the LEVELS operand. LEVELS is only valid with the INDIRECT operand.
PATTERN	Specifies an optional keyword indicating the characters that follow are part of a string.
<i>string</i>	Specifies a string of alphanumeric or DBCS characters. If the pattern string contains blanks, it must be enclosed in single or double quotes. You can further qualify the pattern string using the WORD, PREFIX, or SUFFIX subordinate operands. These subordinate operands describe how the pattern string is to be used.
X' <i>string</i> '	Specifies the hexadecimal string option. Specific unprintable characters may be specified by giving the EBCDIC hexadecimal value. The string must be enclosed in single or double quotes.
T' <i>string</i> '	Specifies the text string option. A character string may be entered regardless of upper or lowercase by using the text option. The string must be enclosed in single or double quotes.
P' <i>string</i> '	Specifies the picture string option. A string profile may be entered instead of exact characters. The string must be enclosed in single or double quotes. Nine special characters are defined by SmartEdit for use in picture strings. These special characters can be combined with other characters: <p>P'=' Any character</p> <p>P'¬' Any nonblank character</p> <p>P'.' Any nondisplay character</p> <p>P'#' Any numeric character</p> <p>P'-' Any non-numeric character</p> <p>P'@' Any alphabetic character (upper or lowercase)</p> <p>P'<' Any lowercase alphabetic character</p> <p>P'>' Any uppercase alphabetic character</p> <p>P'\$' Any special character (not alphabetic or numeric)</p>

Operand	Description
WORD	Specifies the specified pattern string preceded and followed by any non-alphanumeric character (except a hyphen).
PREFIX	Specifies a word that begins with the specified pattern string.
SUFFIX	Specifies a word that ends with the specified pattern string.
Column1	Specifies the column number where the search is to begin.
Column2	Specifies the column number where the search is to end.
Next	Scrolls forward from the current cursor position to the next occurrence of the requested target.
Prev	Scrolls backward from the current cursor position to the previous occurrence of the requested target.
First	Scrolls from the top of the source file to the first occurrence of the requested target.
LAst	Scrolls backward from the bottom of the source file to the first occurrence of the requested target.
All	Scrolls to the first occurrence of the requested target. This is the default value for the SCROLL command.
IN	Restricts the SCROLL command to the specified target type. This operand is optional.

Usage Notes

The screen can also be scrolled by selecting a Search pop-up on the Search pull-down. See ["Search" on page 107](#) for more information.

In Tree View, you can also scroll by selecting View ► Scroll. See [Figure 74 on page 90](#) for more information.

To concatenate targets, place a plus sign (+) between the target names, as shown in this example:

```
IO + CALL
```


These rules apply to concatenation:

- You can use a concatenated dataname wherever a dataname is valid. Dataname subordinate operands (REF, ALIAS, etc.) pertain to all datanames in a concatenated series.
- You can use a concatenated set name wherever a set name is valid. These sets are valid:

LABEL LINE PROGRAM SUBSET

For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, you could qualify it as P120-READ OF VIAFDEM1.

After you enter an FX ZIP-CODE command, you can enter SCROLL PARA PREV with the cursor positioned on an FX command result to see the paragraph name containing the result.

Examples

To scroll to the first occurrence of a line containing the characters ZIP, type this command:

SCROLL ZIP

To scroll to the next highlighted line, type this command:

SCROLL HI NEXT

The SCROLL command supports the ISPF .label function with a target or an IN target.

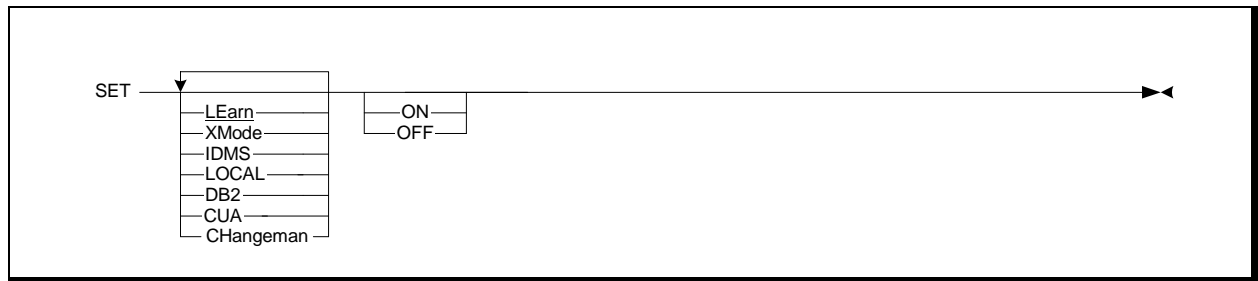
To use .label with a target, type this command:

SC .A

To use .label with an IN target (DATE-CODE), type this command:

SC DATE-CODE IN .A

SET Command



Function

Enables or disables the specified options.

Operands

Operand	Description
Blank	Displays the Options - Processing Modes pop-up that shows the current setting for these operands.
LEarn	Displays all primary commands that are generated internally when actions from the pull-downs are executed. The default for this operand is OFF.
XMode	Specifies that all lines are excluded from the screen before a primary command is executed when this mode is ON. This mode affects all primary commands, with these exceptions: <div style="text-align: center;">BRANCH RFIND SCROLL HIGH</div> The default value for this operand is OFF.
IDMS	Allows SmartEdit to access the IDMS System Catalog to find definitions for IDMS items found in IDMS statements. The default for this operand is OFF.
LOCAL	Allows SmartEdit to access IDMS in Local Mode. The default for this operand is OFF.
DB2	Allows SmartEdit to access the DB2 System Catalog to find definitions for DB2 items found in DB2 statements. The default for this operand is OFF.
CUA	Allows SmartEdit to enable or disable the CUA menu.

Operand	Description
CHangeman	Allows SmartEdit to extract the copy libraries associated with a selected ChangeMan package and automatically populate the SmartEdit copy library screen.
ON	Enables the specified operand.
OFF	Disables the specified operand.

Usage Notes

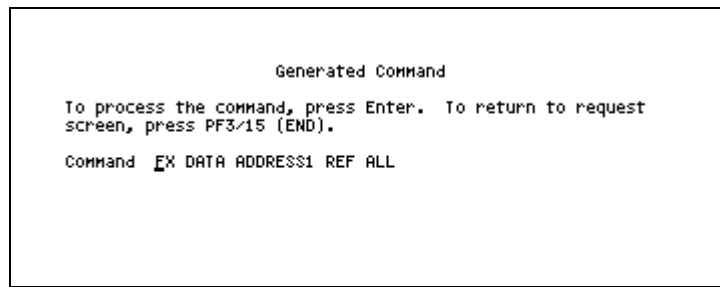
Any of the modes that are enabled or disabled with the SET command can also be enabled or disabled on the Options - Processing Modes pop-up. To display the Options - Processing Modes pop-up, select Options ► Processing Modes or type SET with no operands on any screen.

See ["Options" on page 155](#) for more information about the Options - Processing Modes pop-up.

Generated Command Pop-up

If the LEARN mode has been set to ON using the SET command or the Options - Processing Modes pop-up, the Generated Command pop-up, shown in [Figure 133](#), displays the internally-generated primary command when actions are requested on pop-ups.

Figure 133 • Generated Command Pop-up



Follow the instructions on the pop-up to process or cancel the command.

SWITCH Command

SWItch 

Function

In ISPF 4.1 through 4.8, the SWITCH primary command acts as a toggle to switch between the SmartEdit action bar and the ISPF Editor action bar.

If you are not using ISPF 4.1 through 4.8, the SWITCH command displays the SmartEdit action bar for the current session.

Operands

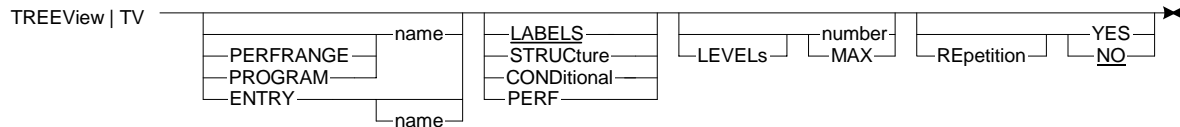
None.

Usage Notes

You can switch between the SmartEdit action bar and the ISPF Editor action bar by using the primary command SWITCH. This acts as a toggle to switch to the other bar.

If the SmartEdit action bar is removed using CUA OFF, the primary command SWITCH displays the ISPF Editor action bar and sets CUA ON.

TREEVIEW Command



Function

Displays the Tree View screen, which shows a structural representation of a program in logical execution order.

Operands

Operand	Description
Blank	Specifies that the default values are used.
PERFRANGE <i>name</i>	Indicates the starting paragraph for the Tree View display. A name is any PROCEDURE DIVISION paragraph name or section name. The PROCEDURE literal or PROC can also specified.
PROGRAM <i>name</i>	Specifies the name of the main program or any nested program representing all the code contained in the program. This includes all the programs physically nested inside the specified program. For COBOL II Release 3 or later programs, you can qualify a data item, label name, or program name that may be ambiguous or used multiple times using OF followed by the program name. For example, if the label P120-READ exists in another program and also in the program VIAFDEM1, you could qualify it as P120-READ OF VIAFDEM1.
ENTRY <i>name</i>	Displays the main program structure and the structure for all ENTRY points if you specify ENTRY without a name. If a name is specified, the structure for that ENTRY point displays.
LABELS	Generates a Tree View display of the structure of the program using the paragraphs and section labels of the control flow statements. LABELS is the default.
STRUCture	Generates a Tree View display of the structure of the program using control flow verbs (PERFORM, GO TO, ALTER, etc.) and program labels.

Operand	Description
CONDitional	Generates a Tree View display of the conditional statements that surround control flow statements, plus the control flow statements and labels.
PERF	Generates a Tree View display of the perform range hierarchy for the program.
LEVELs <i>number</i>	Controls the amount of information displayed on entering Tree View. Specify the number of nested control flow structures to be displayed. The default, 1, displays the first level nested performs.
MAX	Displays the maximum nesting level. However, if the maximum nesting level consists of only repeated perform ranges, then the nested level is truncated to the next higher level that does not consist exclusively of repeated perform ranges.
REPetition	Controls the display of duplicate perform ranges. If YES is specified, all duplicate perform ranges are displayed. If NO is specified, duplicate perform ranges are only flagged as REPEATED, and not expanded unless a ZOOMIN command is entered for the perform ranges. The default is NO.

UPDATE Command



Function

Changes pseudo code lines (created in SmartTest) to actual COBOL source lines, making them part of the program. After you enter the UPDATE command, the File - Generated Code Update Facility pop-up displays. Use this pop-up to specify the program containing pseudo code that is to be updated.

Operands

Operand	Description
Blank	Specifies that the default operands, CHANGE and FIRST, are used.
CHANGE	Updates all pseudo code lines without requesting a confirmation. This is the default.
FIND	Displays pseudo code as note lines in the program being edited, providing a means of viewing changes before actually making them. Note lines include three context lines indicating where the Update facility recommends the lines be placed. The lines to be changed follow the context lines. Execute the UPDATE command again without the FIND operand to insert the new code into the program.
RESET	<p>Displays the File - Generated Code Update Facility pop-up that is used to select a program in an Application Knowledge Repository (AKR) that contains changes to be updated. Enter the AKR and program name(s) on the File - Generated Code Update Facility pop-up, or specify the AKR alone. When you only specify the AKR, the File - Pseudo Code Program Selection pop-up displays for selection of a program.</p> <p>The first time you execute the UPDATE command, the File - Generated Code Update Facility pop-up displays.</p>
Next	Displays or inserts the next block of pseudo code to be updated.
Prev	Displays or inserts the previous block of pseudo code to be updated.

Operand	Description
First	Displays or inserts the first block of pseudo code found in the selected program. This is the default.
LAst	Displays or inserts the last block of pseudo code found in the selected program.
All	Displays or inserts all pseudo code in the selected program.

Usage Notes

You can also display the File - Generated Code Update Facility by selecting File ► Update Pseudo.

The UPDATE command is used to update a source program with temporary COBOL statements created in a SmartTest test session and then stored in an AKR. One use of the UPDATE command is to type UPDATE ALL, select a program, and then press Enter (if sure of all changes). The Update facility opens the AKR, reads all update records, then applies the changes to the source program being edited. All inserted lines are highlighted.

Type UPDATE FIND FIRST to view the changes before they are actually made to the source program. The first block of pseudo code displays as note lines. You can then use RCHANGE to insert these lines as actual code, or RFIND to locate the next block of pseudo code. Note lines are not saved as part of the source file when the editor facility is exited and can be removed from the screen by typing ISPF RESET SPECIAL.

After issuing the UPDATE command, the File - Generated Code Update Facility pop-up displays. On this screen, type the AKR and program name to be updated.

See ["File" on page 77](#) for more information about the File - Generated Code Update Facility pop-up.

Examples

To display the first block of pseudo code to be updated as note lines, type this command:

```
UPDATE FIND FIRST
```

To read all update records and updates the source with the changes, type this command:

```
UPDATE ALL
```


ZOOM COPY Command

ZOOM COPY | ZC 

Function

Brings in all copy members included by the current source module.

Operands

None.

Usage Notes

Copy members can also be included by selecting View ► Zoom All Copies. See ["View" on page 87](#) for more information.

ZOOM DEF Command

ZOOM DEF | ZD

Function

Brings in the definition of any data items on the line corresponding to the current cursor location, or the first line of the screen, including definitions of logical database entities located in the DB2 System Catalog or the IDMS Data Dictionary.

Operands

None. This command is entered in the command input area with no operands and is dependent on the cursor location. This command may also be entered as a line command.


Usage Notes

Definitions of data items can also be displayed by using the ZD line command, or by selecting View ► Zoom Definition. See ["View" on page 87](#) for more information about the View pull-down.

If the cursor is on a data item when ZOOM DEF is issued, only the item displays. If the cursor is anywhere else on the line, all items contained on the line are displayed.

ZOOMIN/ZOOMOUT Commands

ZOOMIn | ZI 

ZOOMOut | ZO 

Function

Displays or excludes source lines according to the hierarchical levels of the program.

Operands

None. These commands are entered in the command input area with no operands and are dependent on the cursor location.

Usage Notes

ZOOMIN and ZOOMOUT are also available on the View pull-down. See ["View" on page 87](#) for more information about the View pull-down.

The ZOOMIN and ZOOMOUT commands show the structure of a program and provide a means of stepping through each level or going directly into or out of a particular section of source code.

Program level hierarchy consists of the IDENTIFICATION DIVISION statement, the PROGRAM-ID statement, and the last physical line of the program. This is especially useful in revealing the hierarchy of nested programs possible with COBOL II Release 3.

PROCEDURE DIVISION hierarchy consists of section labels, paragraph labels, and paragraph code. DATA DIVISION hierarchy consists of sections, FDs, 01 or 77 levels, and all definitions within an 01 level.

Begin by excluding all lines from the screen, then type ZOOMIN. The program level information is shown including the program name. For COBOL II Release 3 modules with sub-programs, this information displays for each of the program names. Type ZOOMIN again. The DIVISION headings are shown. Type ZOOMIN in the command input area and place the cursor on the PROCEDURE DIVISION statement, then press Enter. All PROCEDURE DIVISION headings are displayed. To see the source within a particular heading, type ZOOMIN in the command input area, scroll to the heading, and press Enter. The paragraph names are displayed. Continue this process to see the code and copy statements within several paragraphs. Type ZOOMOUT on a paragraph name to exclude it from the screen once it has been viewed. All source lines under the paragraph name are excluded from the screen.

These are the general guidelines for the ZOOMIN command:

- If the cursor is on an excluded block of lines, ZOOMIN displays the highest structure level within that block of lines. If there are multiple statements at that level, they are also displayed.
- If ZOOMIN is issued repeatedly from the same cursor position, successive heading levels are displayed.

These are the general guidelines for the ZOOMOUT command:

- The cursor cannot be positioned on an excluded line when you type ZOOMOUT.
- ZOOMOUT excludes the levels in a lowest to highest order. If ZOOMOUT is issued repeatedly from the same cursor location, successive heading levels are excluded.

The ZI and ZO line commands can also be used to display or exclude source lines.

Line Commands

Line commands are entered over the line numbers in the prefix area on the screen. Line commands are either single format or block format. Single format refers to a line command being entered on an individual line. [Figure 134](#) shows the line command to exclude line 332.

Figure 134 • Single Format Line Command Example

```
000331      MOVE ZIP-CODE TO HLD-ZIP.
X00332      IF HLD-ZIP-PREFIX EQUAL CUR-PREFIX
000333      NEXT SENTENCE
```

You can include a number with some single format commands to indicate a specified number of lines that are to be processed using the same command. Processing begins with the line on which you enter the command and includes the current line and subsequent lines as indicated by the number specified. If the number of lines exceeds the available source lines, all remaining lines are processed. 99999 is the maximum number you can enter. [Figure 135](#) shows the single format Exclude line command followed by a number to exclude lines 332 through 336.

Figure 135 • Single Format Line Command With a Number Specified

```
000331      MOVE ZIP-CODE TO HLD-ZIP.
X50332      IF HLD-ZIP-PREFIX EQUAL CUR-PREFIX
000333      NEXT SENTENCE
000334      ELSE
000335      PERFORM P150-SUBTOT
000336      THRU P169-EXIT
000337      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX
```

Block format refers to double character line commands that are entered on multiple lines. Block line commands are processed for all lines between (and including) the lines containing the commands. [Figure 136](#) shows the block format Exclude Block command being used to exclude lines 332 through 336.

Figure 136 • Block Format Line Command Example

```
000331      MOVE ZIP-CODE TO HLD-ZIP.
XX0332      IF HLD-ZIP-PREFIX EQUAL CUR-PREFIX
000333      NEXT SENTENCE
000334      ELSE
000335      PERFORM P150-SUBTOT
XX0336      THRU P169-EXIT
000337      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX
```

Remove line commands by typing over them with spaces, by pressing ERASE EOF with the cursor at the beginning of the command, or by typing RESET in the command input area.

Line commands can be ambiguous when entered. [Figure 137](#) shows a common ambiguity.

Figure 137 • Ambiguous Line Command Example

024500	MOVE ZIP-CODE TO HLD-ZIP.
X24600	IF HLD-ZIP-PREFIX EQUAL CUR-PREFIX
024700	NEXT SENTENCE

It is unclear whether two lines, 24 lines, or 246 lines are to be excluded. If two lines are to be excluded, one or more spaces should be entered after the X2. The ERASE EOF key could be used to erase the numbers following the X2.

BR (Branch) Line Command

BR

Function

Positions the cursor at the specified target. This line command is not valid on the Tree View screen.

Operands

None.

Usage Notes

Use the BR (Branch) line command when tracking branching logic. See ["BRANCH Command" on page 192](#) for more information.

DM (DATAMAP) Line Command

DM

Function

Displays the length of a data item and the length and offset for any data items that are contained within the selected data item.

Operands

None.

Usage Notes

Data items are displayed on the Data Item Offset and Length pop-up. See ["View" on page 87](#) for more information.

EC (EDITCOPY) Line Command

EC

Function

Locates a copybook and open it in an edit session using your default editor. If SmartEdit is not your default editor, an ISPF edit session is opened for the specified Copybook. You can use this command on all copybooks (except certain Includes that you can zoom in on) that are currently viewable in the SmartEdit.

The EC line command applies to all copybooks that exist in the Copy/Include libraries you specified on the Options - COPY/Include Libraries screen in SmartEdit.

Operands

None.

Usage Notes

To use the EC line command, follow this step:

- ▶ Type EC in the line command area to the left of the copybook name, place the cursor on the name of the copybook, and press Enter.

These conditions apply for the EC command:

- Panvalet does not allow recursive edits. If you use EC in Panvalet, SmartEdit displays a message indicating the location of the copybook, the dataset source type, and the Source Manager type.
- If you use EC in SCLM, an ISPF edit session is opened and the copybook is no longer under SCLM control.
- If you use EC in Librarian, you must use the ELIPS Librarian Source Manager. If you do not, SmartEdit displays a message indicating the location of the copybook, the dataset source type, and the Source Manager type.
- The EC command is not supported in other source manager editors.

EP (EDITPGM) Line Command

EP

Function

Locates a CALLED program and open it in SmartEdit. You can use this command on all programs that reside in the same library as the current member.

Operands

None.

Usage Notes

To use the EP line command, follow this step:

- ▶ Type EP in the line command area to the left of the program name, place the cursor on the name of the program, and press Enter.

F (First) Line Command

F*n*

Function

Redisplays the specified number of excluded lines. These lines are redisplayed starting with the first line in the block of excluded lines.

Operands

Operand	Description
<i>n</i>	Specifies the number of excluded lines to redisplay. The default is 1. If the number specified is greater than the number of excluded lines, all lines in the excluded block are redisplayed.

Usage Notes

The First line command redisplayes lines excluded as the result of any primary or line command that excludes lines from the screen display.

Example

[Figure 138](#) and [Figure 139](#) show the first two lines in the excluded block being redisplayed.

Figure 138 • F (First) Line Command Example

000002	PROGRAM-ID. PRDEMO.
F2	- - - - - 5 LINES NOT DISPLAYED
000008	INPUT-OUTPUT SECTION.

Figure 139 • F (First) Line Command Results

000002	PROGRAM-ID. PRDEMO.
000003	AUTHOR. WRITTEN BY ASG AT LANGLVL 2.
000004	ENVIRONMENT DIVISION.
- - -	- - - - - 3 LINES NOT DISPLAYED
000008	INPUT-OUTPUT SECTION.

J (Jump) Line Command

J

Function

Returns to the editor screen at the location corresponding to the line where this command is issued on the Tree View screen. This line command is valid only from the Tree View screen.

Operands

None.

L (Last) Line Command

`Ln`

Function

Redisplays the specified number of excluded lines. These lines are redisplayed starting with the last line in the block of excluded lines.

Operands

Operand	Description
<i>n</i>	Specifies the number of excluded lines to redisplay. The default is 1. If the number specified is greater than the number of excluded lines, all lines in the excluded block are redisplayed.

Usage Notes

The Last line command redisplay lines excluded as the result of any primary or line command that excludes lines from the screen display.

Example

[Figure 140](#) and [Figure 141](#) show the Last line command being used to redisplay the last two lines in the excluded block.

Figure 140 • L (Last) Line Command Example

```
000002  PROGRAM-ID. VIAIDEMO.
L2      - - - - - 5 LINES NOT DISPLAYED
000008  INPUT-OUTPUT SECTION.
```

Figure 141 • L (Last) Line Command Results

```
000002  PROGRAM-ID. VIAIDEMO.
- - - - - 3 LINES NOT DISPLAYED
000006  SOURCE-COMPUTER.  IBM-370.
000007  OBJECT-COMPUTER.  IBM-370.
000008  INPUT-OUTPUT SECTION.
```

S (Show) Line Command

S_n

Function

Redisplays the specified number of excluded lines. These lines are redisplayed starting with the first line in the excluded block.

Operands

Operand	Description
n	Specifies the number of consecutive excluded lines to show. The default is 1. If the number specified is greater than the number of excluded lines, all lines in the excluded block are redisplayed.

Usage Notes

The Show line command redisplay lines excluded as the result of any primary or line command that excludes lines from the screen display.

Example

[Figure 142](#) and [Figure 143](#) show the Show line command being used to redisplay two excluded source lines.

Figure 142 • S (Show) Line Command Example

```
000002  PROGRAM-ID. VIAIDEMO.
S2      - - - - - 5 LINES NOT DISPLAYED
000008  INPUT-OUTPUT SECTION.
```

Figure 143 • S (Show) Line Command Results

```
000002  PROGRAM-ID. VIAIDEMO.
000003  AUTHOR. WRITTEN BY ASG AT LANGLVL 2.
000004  ENVIRONMENT DIVISION.
- - - - - 3 LINES NOT DISPLAYED
000008  INPUT-OUTPUT SECTION.
```

SS (Show Block) Line Command

SS

Function

Redisplays a block of excluded lines.

Operands

None.

Usage Notes

The Show Block line command is entered in the line command area on the first and last line of the excluded blocks to be redisplayed.

Example

[Figure 144](#) and [Figure 145](#) show the Show Block line command being used to redisplay two blocks of excluded lines.

Figure 144 • SS (Show Block) Line Command Example

```
000401  C4H-TO-NUM.
SS0402  * MOVE COMP-4 (HALFWD) TO NUMERIC (UNSIGNED)
- - - - - 4 LINES NOT DISPLAYED
000407  * MOVE COMP-4 (HALFWD) TO NUMERIC (SIGNED)
- - - - - 4 LINES NOT DISPLAYED
SS0412  C4H-TO-C1.
```

Figure 145 • SS (Show Block) Line Command Results

```
000401  C4H-TO-NUM.
000402  * MOVE COMP-4 (HALFWD) TO NUMERIC (UNSIGNED)
000403      MOVE N-C4HALF  TO 77-NU.
000404      MOVE N-C4HALFS TO 77-NU.
000405      MOVE X-C4HALF  TO 77-NU.
000406      MOVE X-C4HALFS TO 77-NU.
000407  * MOVE COMP-4 (HALFWD) TO NUMERIC (SIGNED)
000408      MOVE N-C4HALF  TO 77-NUS.
000409      MOVE N-C4HALFS TO 77-NUS.
000410      MOVE X-C4HALF  TO 77-NUS.
000411      MOVE X-C4HALFS TO 77-NUS.
000412  C4H-TO-C1.
```

X (Exclude) Line Command

Xn

Function

Excludes a line or group of lines from being displayed. Excluded lines are shown as a row of dashes with the number of excluded lines indicated.

Operands

Operand	Description
n	Specifies the number of consecutive lines to exclude. The default is 1. If the number specified is greater than the number of available lines, all remaining lines are excluded.

Usage Notes

Excluded lines are shown as a row of dashes with the number of excluded lines indicated.

Example

[Figure 146](#) and [Figure 147](#) show the Exclude line command being used to exclude lines 332-336.

Figure 146 • X (Exclude) Line Command Example

```
000331      MOVE ZIP-CODE TO HLD-ZIP.
X5 332      IF HLD-ZIP-PREFIX EQUAL CUR-PREFIX
000333          NEXT SENTENCE
000334      ELSE
000335          PERFORM P150-SUBTOT
000336              THRU P169-EXIT
000337      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX
```

Figure 147 • X (Exclude) Line Command Results

```
000331      MOVE ZIP-CODE TO HLD-ZIP.
- - - - - 5 LINES NOT DISPLAYED
000337      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX
```

XX (Exclude Block) Line Command

XX

Function

Excludes a block of lines from being displayed. Excluded lines are shown as a row of dashes with the number of excluded lines indicated.

Operands

None.

Usage Notes

The Exclude Block line command is entered in the line command area on the first and last line of the block of lines to be excluded from being displayed.

Examples

[Figure 148](#) and [Figure 149](#) show the Exclude Block line command being used to exclude a block of five lines.

Figure 148 • XX (Exclude Block) Line Command Example

```
000331      MOVE ZIP-CODE TO HLD-ZIP.
XX0332      IF HLD-ZIP-PREFIX EQUAL CUR-PREFIX
000333          NEXT SENTENCE
000334      ELSE
000335          PERFORM P150-SUBTOT
XX0336          THRU P169-EXIT
000337      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX
```

Figure 149 • XX (Exclude Block) Line Command Results

```
000331      MOVE ZIP-CODE TO HLD-ZIP.
- - - - - 5 LINES NOT DISPLAYED
000337      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX
```


ZD (Zoom Def) Line Command

ZD

Function

Brings in the definition of any data items on the line where ZD was entered. This command is not valid on the Tree View screen, or outside the PROCEDURE DIVISION.

Operands

None.

Usage Notes

Data definitions brought in by ZD (Zoom Def) include definitions of logical database entities located in the DB2 System Catalog or the IDMS Data Dictionary.

See ["ZOOM DEF Command" on page 280](#) for more information.

ZI (Zoom In) Line Command

ZI

Function

Redisplays excluded source lines according to the hierarchical levels in the program and to expand copy members in context. The ZI line command is used in conjunction with the ZO (Zoom Out) line command to show the structure of a program and provide a means of stepping through each level or going directly into or out of a particular section of source code.

Operands

None.

Usage Notes

The ZI (Zoom In) line command is used to show the structure of a program. PROCEDURE DIVISION hierarchy consists of section labels, paragraph labels, and paragraph code. DATA DIVISION hierarchy consists of sections, FDs, 01 or 77 levels, and all definitions within an 01 level.

If the ZI (Zoom In) line command is entered on an excluded block of lines, the highest structure level within that block of lines displays. If there are multiple statements at that level, they are also displayed.

See ["ZOOMIN/ZOOMOUT Commands" on page 281](#) for more information about displaying and excluding source lines according to the hierarchical levels of the program.

ZO (Zoom Out) Line Command

ZO

Function

Redisplay a source line in its original format after a zoom line command has been entered. The ZO (Zoom Out) line command is used in conjunction with the ZI (Zoom In) line command to show the structure of a program and provide a means of stepping through each level or going directly into or out of a particular section of source code. The ZI (Zoom In) and ZO (Zoom Out) line commands are used to display or exclude source lines according to the hierarchical levels of the program.

Operands

None.

Usage Notes

The ZO (Zoom Out) line command can only be entered on a line that has been displayed using the ZI (Zoom In) line command.

See ["ZOOMIN/ZOOMOUT Commands" on page 281](#) for more information about displaying and excluding source lines according to the hierarchical levels of the program.

The ZO (Zoom Out) line command can also be used to remove lines that were displayed in a window on the screen, as the result of a ZI (Zoom In) line command.

12

Command Quick Reference

All SmartEdit primary commands are listed alphabetically for quick and easy reference.

Command	Abbreviation	Default PF Key	Description
& (Retain)	none	none	Executes a primary command and keeps it displayed in the command input area for additional use.
ACTION	ACT	none	Initiates the cursor sensitive function previously selected from a pull-down.
ALLOCDEF	ADEF	none	Displays the Options - Product Allocations pop-up.
BRANCH	B	none	Positions the cursor at the specified target.
CHECK	none	none	Provides an online syntax check mechanism for the COBOL source currently being edited.
COBEDIT	none	none	Activates and deactivates the COBOL editor under ISPF/PDF Edit.
CUA	none	none	Activates and deactivates the CUA interface for SmartEdit.
DISPLAY	DIS	none	Excludes from the screen all statements in the program that are not contained in the current logic segment. This command is valid <i>only</i> if Encore is installed.
EQUATE	EQ	none	Defines a name for a character string.

Command	Abbreviation	Default PF Key	Description
EXCLUDE	X EX	none	Performs a FINDXTND command on the specified target, excluding the resulting lines from display.
FINDXTND	FX FINDX	none	Performs a COBOL intelligent search of the source code for one or all occurrences of the specified target.
HELP	H ?	PF01/13	Displays information about the current screen, commands, error messages, or abend codes.
HIGH	HI	none	Highlights source code lines containing the specified targets.
JUMP	none	none	Returns to the editor screen at the location corresponding to the cursor position on the Tree View screen.
LEVELS	LEVEL	none	Redisplays the current Tree View screen to the level specified.
LIST	LI	none	Displays the specified list screen that contains pertinent information for the type of list requested (e.g., CALLs, equates, etc.).
LPRINT	LP	none	Copies the lines that contain the specified target to the List file.
LPUNCH	LPU	none	Copies the lines that contain the specified target to the Punch file.
PARMDEF	PDEF	none	Displays the Options - Product Parameters pop-up.
PREF	PRE	none	Displays the View - Paragraph Cross Reference pop-up for the requested target.
PREPROC	none	none	Displays the Options - Preprocessors pop-up that is used to select a preprocessor group.

Command	Abbreviation	Default PF Key	Description
PRINTLOG	PLO PLOG	none	Displays the Options - Log/List/Punch Definition pop-up that is used to process the Log, List, and Punch files.
PRINTLST	PLI PLIST	none	Displays the Options - Log/List/Punch Definition pop-up that is used to process the Log, List, and Punch files.
PRODLVL	none	none	Displays the current SmartEdit and Center product and release level.
RECALL	REC	none	Displays the previous primary command or message.
REDO	RED	none	Executes the corresponding repeat command from the cursor position.
REFRESH	REF	none	Displays the Edit options screen to allow the copy library names to be updated and then brings in current copies of all of the copy members.
REPEAT	REPE	none	Executes the last stacked ESW primary command again.
RESET	RES	none	Removes highlighting, erases tags in columns 1 through 6 or 73 through 80, and redisplayes excluded lines.
RFIND	none	PF05/17	Repeats the last FIND or FINDXTND command from the cursor position.
RHIGH	RHI	none	Repeats the last HIGH command from the cursor position.
RPREF	RP	none	Redisplays the last View - Paragraph Cross Reference pop-up.
RSCROLL	RSC	none	Repeats the last SCROLL command from the cursor position.
RTREEVW	RTV RTREE	none	Returns to the previous Tree View screen.
SCROLL	SC	none	Positions the screen to the first line that contains the specified target.

Command	Abbreviation	Default PF Key	Description
SET	none	none	Enables or disables the mode indicated by the specified operand.
SWITCH	SW	none	In ISPF 4.1 through 4.8, the SWITCH primary command acts as a toggle to switch between the SmartEdit action bar and the ISPF Editor action bar.
TREEVIEW	TV	none	Displays the Tree View screen, which shows a structural representation of a program in logical execution order.
UPDATE	UPD	none	Changes pseudo code lines (created in SmartTest) to actual COBOL source lines, making them part of the program.
ZOOM COPY	ZC	none	Brings in all copy members included by a current source module.
ZOOM DEF	ZD	none	Brings in the definition of any data-items on the line corresponding to the cursor position, or the first line of the screen, including definitions of logical database entities located in the DB2 System Catalog or the IDMS Data Dictionary.
ZOOMIN	ZOOMI ZI	none	Displays source lines according to the hierarchical levels of the program.
ZOOMOUT	ZOOMO ZO	none	Excludes source lines according to the hierarchical levels of the program.

13

Help Facility

This chapter describes the Help Facility and contains these sections:

Section	Page
Introduction	303
Help Navigational Commands	305
Screen and Pop-up Help	306
Command Help	307
General Information	309
Specific Information	310
Help Abends	311
Help Messages	312

Introduction

Comprehensive and context-sensitive help facilities are provided that answer most questions online. The Help Tutorial contains help information for several different subjects, such as screens, pop-ups, commands, messages, and abends. The Help Tutorial also includes a Table of Contents that lists general information subjects, and a comprehensive Index for viewing specific information.

You can reach the SmartEdit online help facilities using several means. Selecting Help on the action bar displays the Help pull-down, which is described fully in ["Help" on page 175](#). Help can also be requested by pressing PF01/PF13, or by typing HELP or ? (question mark) in the command input area on any screen or pop-up.

These are the various online help information, and the means of accessing them, provided by SmartEdit:

Help Topic	How to Access
Screen or pop-up help	Provides help for the current screen or pop-up is requested by typing HELP , pressing PF1/13 , or selecting Current Screen on the Help pull-down. No messages can display on the screen or pop-up at the time this help is requested.
Command help	Provides help for a command is requested by typing the command in the command input area and pressing PF01/13 , typing HELP COMMANDS , typing HELP with the command name specified as the operand, or by selecting All Commands or Specific Command on the Help pull-down.
General Information	Provides general help information is requested by issuing the TOC command from within the Help Tutorial , or by selecting Help ► Table of contents .
Specific information	Provides help for specific topics is requested by typing INDEX from within the Help Tutorial , by selecting the appropriate Specific Message or Command option from the Help Table of Contents , or by selecting Help ► Index . You can view help for a specific topic by selecting the appropriate index entry.
Abends	Provides help for ESW user abends is requested by typing HELP ABENDS , or selecting Common Abends on the Help pull-down. The Abends screen displays. Select Topic 2 on this screen to display the ASG Abend Codes screen, which lists all the ESW user abends, and explanations for each abend.
Messages	Provides help for a short message, displayed in the upper right corner of the screen, is requested by typing HELP or by pressing PF1/13 . The corresponding long message displays. You can display help for a specific long message by typing HELP plus the msg# operand, or by selecting Help ► Current Message or Help ► Specific Message .

Help Navigational Commands

All of the online help topics listed in the previous table are contained in the Help Tutorial. You can reach each online help topic from anywhere within the Help Tutorial using the Help Table of Contents or Index. After you access the Help Tutorial, these commands are available for navigating within the Help Tutorial:

Help Command	Purpose
BACK	Redisplays the previous Help Tutorial screen.
END	Exits the Help tutorial.
ENTER	Displays the next screen in a continuation series.
INDEX	Displays the first screen of the Help Index.
SKIP	Goes directly to the next subject.
TOC	Displays the Help Table of Contents.
UP	Displays the next higher level subject.
Alpha character	Displays the Index screen corresponding to the alphabetic character you enter.

Screen and Pop-up Help

Help for the current screen or pop-up is accessed by typing HELP, by pressing PF1/13 with no messages displaying on the screen, by typing HELP SCREEN, or by selecting Current Screen on the Help pull-down. The Help Tutorial for the current screen or pop-up, as shown in [Figure 150](#), displays.

The Help Tutorial for each screen or pop-up describes all the options available on that screen or pop-up, lists descriptions of all the screen fields, and notes any special processing considerations.

Figure 150 • Screen and Pop-up Example

```

----- Options - Product Parameters ----- HELP
===> _

Use the Options - Product Parameters pop-up to set database support parameters
that affect the online operation of ASG-SmartEdit. This pop-up displays by
selecting Options > Product Parameters, or by typing PARMDEF on any screen.

Field Descriptions

Alarm      Enter YES or NO. If YES, the audible alarm on the terminal sounds
           when an error message displays. The default is YES.

Cursor Character
           Sets the cursor token substitution character. Cursor substitution
           saves time by allowing you to use this character in a command and
           then place the cursor under the token in the source code.
           % (percent) is the default.

                                           (More...press Enter to continue)

```

Command Help

Help for a specific command is requested by typing the command in the command input area and pressing PF1/13, typing `HELP` followed by the desired command name, or by selecting `Help ► Specific Command`. A long message displays describing the command. Pressing PF1/13 again displays the Help Tutorial screen for that command, as shown in [Figure 151](#). The Help Tutorial for each command displays the command syntax diagram, and gives a description of each operand in the command. Typing UP on a command help screen displays a complete list of all SmartEdit primary commands.

For help on all SmartEdit commands, type `HELP COMMANDS`, or select `All Commands` on the Help pull-down. This displays a complete list of all SmartEdit commands. By selecting the appropriate number from this list, you can display information about a particular command.

Figure 151 • Command Help Example

```

ASG-SmartEdit      ----- RECALL ----- HEL
===> _

The RECALL command displays the previous command or message. The last twenty
commands that have been executed and the last twenty messages that have been
displayed are stacked.

The RECALL command syntax is:
RECALL ----->
      | -COMMAND| CMD- |      | -NEXT- |
      | -MESSAGE| MSG- |      | -PREV- |

                                Minimum Abbreviations are in CAPS
                                Default operands are highlighted
                                LEGEND: ---required----->
                                           | -optional- |

The following topic will be presented only if explicitly selected by number:
  1 - Operand Descriptions
----- Cur panel = UFTREC   Prev panel = USPEDDE2 Last msg = ISR2000 -----

```

For some commands on the editor screen only, after the long message displays, pressing PF1/13 displays NOTES giving specific examples for using the command, as shown in [Figure 152](#). Once the NOTES are displayed, pressing PF1/13 again displays the Help Tutorial for that command. The Help NOTES displays for those commands where specific examples are helpful.

Figure 152 • Help NOTES Example

```

ASG-SmartEdit ----- ASG.VIACENxx.CNTL(VIAFDEMO) - 01.03 ----COLUMNS 007 072
COMMAND ==>                                SCROLL ==> PAGE
ASG-4723I RECALL REDISPLAYS THE PREVIOUS COMMAND OR MESSAGE.
=NOTE= +----- Examples ----- RECALL ----- Descriptions -----+
=NOTE= | RECALL                                | Recalls the last command that was |
=NOTE= |                                     | entered in the primary command   |
=NOTE= |                                     | area.                            |
=NOTE= | REC MSG                               | Recalls the last message that was |
=NOTE= |                                     | displayed in the message area.   |
=NOTE= | RECALL NEXT                           | Once recalling is started, NEXT   |
=NOTE= |                                     | may be used to reverse its       |
=NOTE= |                                     | direction.                        |
=NOTE= +-----+-----+-----+-----+
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. VIAFDEMO.
000300 AUTHOR. ASG AT LANGLVL 2.
000400 ENVIRONMENT DIVISION.
000500 CONFIGURATION SECTION.
000600 SOURCE-COMPUTER. IBM-370.
000700 OBJECT-COMPUTER. IBM-370.
000800 INPUT-OUTPUT SECTION.
000900 FILE-CONTROL.
001000         SELECT MASTERIN ASSIGN TO S-MASTERIN.

```

To exit from Help NOTES before displaying the Help Tutorial, issue the RESET command.

General Information

General help information is requested by entering the Help Tutorial, then issuing the TOC command, or by selecting Help ► Table of contents. The Help Table of Contents screen, as shown in [Figure 153](#), displays.

Figure 153 • Help Table of Contents Screen

```

ASG-ESW - COBOL Editing R7.0  Help Table of Contents  ----- HELP
===> _

The topics below represent general categories of information about the ASG-ESW
COBOL Editing component, ASG-SmartEdit. To get help for a pull-down, select
the Action Bar topic. This Help Table of Contents can be redisplayed from
any help screen by entering the TOC command.

These topics are presented only if explicitly selected by number:

  1 - Overview of ASG-SmartEdit
  2 - Introduction to CUA
  3 - The Action Bar
  4 - Release 6.0 Summary of Revisions
  5 - Getting Started with ASG-SmartEdit
  6 - Treeview Facility
  7 - Help Information
  8 - Exiting ASG-SmartEdit
  9 - Customer Support
 10 - Frequently Asked Questions
 11 - Index for ASG-SmartEdit Help

----- Cur panel = VFITOC  Prev panel = USPEDDE2 Last msg = ISR2000 -----

```

Specific Information

Help for specific topics is requested by typing INDEX from within the Help Tutorial, by selecting option 10 on the Help Table of Contents, or by selecting Help ► Index. Help for a specific topic can then be viewed by selecting the appropriate Index entry.

On any Index screen, typing an alphabetic character displays the Index screen corresponding to that character, as shown in [Figure 154](#).

Figure 154 • Help Index Example

```

ASG-ESM - COBOL Editing R7.0 ----- Index -----HELP
===> _

All of the major topics discussed in HELP are in this index. The index
selection screens list topics in alphabetical order. You can page through
the index by pressing Enter, or go directly to an index screen by entering
the first letter of the subject.

Select a topic from the index selection screen by entering the two- or
three-character option preceding the desired topic.

This index can be displayed from any HELP screen by typing INDEX or I in the
command input area.

----- Cur panel = VFTIND   Prev panel = USPEDDE2 Last msg = ISR2000 -----

```


Help Abends

Help for ESW user abends is requested by typing `HELP ABENDS`, typing `ABENDS` in the command input area and pressing `PF1/13`, or by selecting Common Abends on the Help pull-down. The `ABENDS` screen displays. Selecting Topic 2 on this screen displays the ASG Abend Codes screen, shown in [Figure 155](#), which lists all the ESW userabend messages, and explanations for each message.

Figure 155 • ASG Abend Codes Screen

```

ASG-ESW - COBOL Editing R7.0 -- ASG Abend Codes ----- HELP
==>
Abend codes in the range 900 - 999 (X'384 - X'3E7') bypass ESW error
recovery, causing theabend to be handled by ISPF or by the system. If
the problem cannot be resolved, contact ASG Customer Support.

965 X'3C5'    Unable to intercept program.
967 X'3C7'    The ASG-Center AUTHORIZE password was not specified during
              installation.
968 X'3C8'    An internal error occurred during initialization.
970 X'3CA'    A package load module was CALLED directly.
972 X'3CC'    The ESW Edit Monitor encountered a severe error.
974 X'3CE'    An invalid VIASBASE module was found. The current product
              expects a level of CE050 or greater. Enter HELP 4988 for more
              information.

                      (More...press Enter to continue)
----- Cur panel = USTABND0 Prev panel = USTABND  Last msg = ISR2000 -----

```

Help Messages

SmartEdit messages are displayed in the long message area, which is the third line on the screen. The message format is:

ASGnnnnx text

where

nnnn is the message number.

x is one of the severity levels listed below.

text is the long or short message text.

This table describes the message severity levels:

Level	Type	Description
I	Informational	Indicates that no action required.
W	Warning	Indicates that an error condition exists that is not critical.
E	Error	Indicates that a critical error condition exists.
D	Disaster	Indicates that a serious error condition exists; the product is unable to continue.
T	Termination	Indicates that the product terminated with the specified error.

Short messages are displayed when available. Long messages are displayed if a short message does not exist, or when help is requested immediately following a displayed short message.

Help for a specific message displays by selecting Help ► Current Message, Help ► Specific Message, or by typing HELP followed by the message number. The Help Explanation and Action screen for that message displays, as shown in [Figure 156](#).

Figure 156 • Help Explanation and Action Screen

```

                                HELP Explanation and Action Panel
Command ==> _____ Scroll ==> 3

Additional support may be found at our Web Site: support.asg.com
ASG00650  END OF PROGRAM; USE BRANCH BACKUP TO FOLLOW OTHER BRANCHES.

EXPLANATION:
  This is a warning message indicating that BRANCH has reached the
  physical end of the program.

ACTION:
  If you wish to branch to other paths, use BRANCH BACKUP to reach
  the desired decision point and then use BRANCH to follow another
  path.

***** BOTTOM OF DATA *****

```

Printing Messages

Print all SmartEdit messages or a range of messages using the VIASMPRT program. The VIASMPRT program produces a listing of the specified messages that includes:

- Message number
- Short message (if available)
- Long message
- Explanation of the message
- Action (if any)

JCL to execute the VIASMPRT program is in ASG.VIACENxx.CNTL(VIASMPRT). The entire message file prints unless a specific range is specified in the PRM parameter. For example, PRM='START=300,END=499' would print messages 300 through 499.

You can specify the ALL keyword in the PRM parameter to print all messages.

The default value for START is 1; the default value for END is 5000. If only the START value is entered, messages print starting at the message number specified and ending with 5000. If only the END value is entered, messages print starting with 1 and ending with the message number specified.

The NOTES keyword specifies that any notes associated with a message be printed. The default is NONOTES. Typically, notes are provided to show Center primary commands.

Use the VIASMPRT JCL, shown in [Figure 157](#), to print the ESW messages.

Figure 157 • JCL for Printing Messages

```

//ASG JOB ( ),'ASG-CENTER VIASMPRT'
//* INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
//* *****
//* * ASG, INC. ASG-CENTER *
//* * * * *
//* * UTILITY TO PRINT ASG MESSAGES *
//* * * * *
//* *****
//*
//VIASMPRT PROC VIASOFT='ASG', ASG HI-LVL NODES
// CENTER='VIACENXX', ASG MIDDLE NODES
// SYSOUT='*', PRINT OUTPUT MESSAGE CLASS
// PRM=' ' PARM FOR MESSAGES TO BE PRINTED
//*
//* *****
//* * * * *
//* * MESSAGE PRINT UTILITY *
//* * * * *
//* * THIS PROGRAM WILL PRINT ALL OF THE MESSAGES IN THE ASG *
//* * MESSAGE FILE AND THE HELP TEXT ASSOCIATED WITH EACH *
//* * MESSAGE IT WILL PRINT THE ENTIRE FILE BY DEFAULT. YOU MAY *
//* * SELECT A GIVEN RANGE OF MESSAGES BY SPECIFYING THE OPTION- *
//* * AL PARAMETER KEYWORDS: START AND END. FOR EXAMPLE: *
//* * PRM='START=300,END=499' *
//* * WILL PRINT MESSAGES NUMBER 300 THROUGH 499, INCLUSIVE. *
//* * THE DEFAULT VALUES FOR START AND END ARE 1 AND 99999 *
//* * RESPECTIVELY. CONSEQUENTLY THE PRM VALUE 'END=300' WILL *
//* * PRINT MESSAGES 1 THROUGH 300, AND THE PRM VALUE *
//* * 'START=4000' WILL PRINT MESSAGES 4000 THROUGH 99999. *
//* * * * *
//* * AN OPTIONAL KEYWORD, NOTES, WILL ALSO PRINT ANY NOTES *
//* * ASSOCIATED WITH A MESSAGE. *
//* * * * *
//* * ADDITIONALLY, THE KEYWORD 'ALL' WILL EXPLICITLY PRINT ALL *
//* * MESSAGES. *
//* *****
//*
//*
//VIAMPRT EXEC PGM=VIASMPRT,REGION=4096K,
// PARM='&PRM'
//STEPLIB DD DSN=&VIASOFT..&CENTER..LOADLIB,DISP=SHR
//VIAMSGS DD DSN=&VIASOFT..&CENTER..VIAMSGS,DISP=SHR
//SYSPRINT DD SYSOUT=&SYSOUT
//VIAPRINT DD SYSOUT=&SYSOUT
//VIALOG DD SYSOUT=&SYSOUT
//SYSUDUMP DD SYSOUT=&SYSOUT
//*
// PEND
//*
//*
//VIASMPRT EXEC VIASMPRT PRINT MESSAGES
//*

```

[Figure 158](#) shows the VIASMPRT JCL output from the job.

Figure 158 • VIASMPRT Output

```
PRINTING MESSAGES FROM 1051 TO 1052.
2 MESSAGES PRINTED.
END OF MESSAGE PRINT PROCESSING.
ASG1051  APPLICATION KNOWLEDGE REPOSITORY (AKR) MEMBER DIRECTORY IS FULL.

      EXPLANATION:
        The AKR Directory is full.

      ACTION:
        Insufficient space is left on the directory.  Expand the AKR by
        using the 3.0 screen (AKR Utilities).

ASG1052  APPLICATION KNOWLEDGE REPOSITORY (AKR) MEMBER NOT FOUND.

      EXPLANATION:
        The specified program was not found in the specified Application
        Knowledge Repository.

      ACTION:
        Verify the spelling of the member name and that it exists in the
        AKR specified, perhaps the program resides in another AKR.
```

Glossary

action bar

The line of keywords at the top of a screen. Each keyword represents a category of actions that may be performed on that screen. An action is selected by moving the cursor to the desired keyword and pressing Enter. See ["Product Overview" on page 25](#) for more information.

COBOL subset

COBOL verbs of a similar nature that have been grouped together. For example, READ, WRITE, OPEN, CLOSE are grouped into the IO subset. See ["Product Overview" on page 25](#) for more information.

command

SmartEdit accepts two types of ESW commands: primary commands and line commands.

command input area

The field on SmartEdit screens where primary commands are entered, indicated by ===> on the fourth line of a screen or the second line of a pop-up.

control flow

Describes how execution can transfer from one COBOL statement to another.

cursor position

The current location of the cursor on the screen, used as a starting point in certain commands.

cursor substitution character

A facility used to place a substitution character in commands. The command and cursor substitution character are typed in the command input area, the cursor is placed on the desired token, then Enter is pressed. SmartEdit locates the cursor and reads the specified token as part of the command. The cursor character is set on the Options - Product Parameters pop-up. See ["Options" on page 155](#) for more information.

data flow

Describes the path of executable statements leading to how the value of datanames are tested, used, and changed.

dataname

The standard COBOL term for fields defined in the DATA DIVISION of a COBOL program. Variable names, files, groups, array elements, and fully-qualified datanames. Multiple datanames can be searched for at once using a plus sign (+), for example:

```
FX ZIP-CODE + HLD-ZIP
```

data usage

Defines how a data item is used. A data item can be used in one of four ways: 1 USE - when the value is used or tested, 2 MOD - when the value is set or changed (modified), 3 DEF - indicates the statements in the DATA DIVISION where it is defined, 4 REF - means any one of the above.

DBCS

See [Double Byte Character Set \(DBCS\)](#).

Double Byte Character Set (DBCS)

A character set that uses two bytes to represent each character. Various Double Byte Character Sets are used with languages such as Chinese and Japanese that cannot be represented with single byte codes.

equate

A substitution name for a character string. The substitution name is created using the EQUATE command or the Options -Equate pop-up. Long commands, patterns, datanames, etc. can be equated to a substitution name.

help

SmartEdit Help has three levels: Long messages, NOTES, and tutorial screens. Specific command information is available by entering the command and pressing PF01/13. Explanations and actions for messages are available by requesting help for the desired message. Help may also be called from any SmartEdit screen. See ["Help" on page 175](#) for more information.

label name

Any paragraph or section name of the PROCEDURE DIVISION, as well as the literals PROCEDURE and PROC. Label name specifies all transfers of control to a paragraph or section.

line commands

Commands that affect specific lines or blocks of lines, entered in the line command area (columns 1 through 6 of the editor screen).

list file

A file that is allocated the first time the LPRINT command is executed. This file is used for LPRINT output. See the Options - Log/List/Punch Definition pop-up in [Figure 121 on page 165](#) for more information.

live exit

An abnormality in program control caused by out of perform range GO TOs and overlapping perform ranges.

log file

A file that is allocated by SmartEdit and used for error messages and log commands. See the Options - Log/List/Punch Definition pop-up in [Figure 121 on page 165](#) for more information.

long message

A diagnostic or error message that displays on line five of a screen or line three of a pop-up. A long message is sometimes preceded by a short message that displays in the upper right corner of the screen. If the short message does not provide enough information, press PF1/13 to display the long message.

MLE

Millennium Language Extension.

online help

See [help](#).

paragraph name

Any PROCEDURE DIVISION paragraph or section name, and the PROCEDURE and PROC literals. Paragraph name includes the entire paragraph or section.

path

Any group of executable statements that describe possible execution flows of the COBOL program.

perform range

A perform range consists of the source code contained in a PERFORM statement, and includes all code that is executed as a result of GO TOs, PERFORMs, etc. within that PERFORM.

pop-up

A window that displays as the result of selecting an item on a pull-down or pop-up, or as the result of entering certain commands. It is superimposed on the screen to allow entry of information for the requested action. See ["Product Overview" on page 25](#) for more information.

primary command

An instruction entered in the command input area of the screen.

program

The program source member name, the name specified in the IDENTIFICATION DIVISION of a COBOL program, or the CSECT name of a program that is not COBOL.

pull-down

The list that displays when an action is selected on the action bar. On a pull-down, actions followed by an ellipsis (...) display a pop-up when selected. Actions not followed by an ellipsis (...) immediately activate internal commands. See ["Product Overview" on page 25](#) for more information.

punch file

A file that is allocated the first time the LPUNCH command is executed. This file is used for LPUNCH output. See the Options - Log/List/Punch Definition pop-up in [Figure 121 on page 165](#) for more information.

recursion

A perform range or paragraph that performs itself.

SBCS

See [Single Byte Character Set \(SBCS\)](#).

screen

A full-width display of information containing an action bar as the first line. SmartEdit screens are modeled after TSO/ISPF screens. See ["Product Overview" on page 25](#) for more information.

screen subsets

Lines that have been acted upon by an interactive command that has caused them to be in one of the following screen display set types:

Highlighted | HI Excluded | X

NONHighlighted | NHI NONExcluded | NX

See ["Product Overview" on page 25](#) for more information.

short message

A diagnostic or error message that displays in the upper right corner of a screen. A short message is sometimes followed by a long message that displays on line five of a screen and line three of a pop-up. If the short message does not provide enough information, press PF1/13 to display the long message.

Single Byte Character Set (SBCS)

A character set that uses one byte to represent each character. Single Byte Character Sets are used with languages such as English where the characters can be represented with a one-byte code.

Storage Management Subsystem (SMS)

An operating environment that automates and centralizes the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

subset

A COBOL subset or screen subset. See ["Product Overview" on page 5](#) for more information.

target

The object of a SmartEdit primary command.

work file

A file that is allocated upon entry into SmartEdit. See the Options - Product Allocations pop-up in [Figure 121 on page 165](#) for more information.

Symbols

& Retain command [185, 189](#)
+ Tree View tag [94](#)

A

abbreviations for commands [299](#)
abend help [311](#)
action bar
 description [5](#)
 displaying [35](#)
 shortened [6](#)
ACTION command [89](#)
activating SmartEdit [23](#)
alarm [158](#)
alias names [214](#)
Alliance
 accessing from ESW screen [xvi](#)
 description [xiii](#)
 linking [xvi](#)
ALLOCDEF command [139, 191](#)
ASG Abend Codes screen [311](#)
AutoChange
 accessing from ESW screen [xvi](#)
 description [xiii](#)

B

BR (Branch) line command [284](#)
BRANCH BACKUP command [39](#)
BRANCH command [95, 109, 192](#)
branching logic [35, 193](#)
Bridge
 accessing from ESW screen [xvi](#)
 description [xiii](#)

C

CALL statements, locating [51](#)
CALLs [148](#)
Center
 product level [253](#)
Center, description [xiii](#)
ChangeMan processing mode [170](#)
character string [204](#)

character substitution [72, 186](#)
Check - Compiler Options pop-up
 example [140](#)
 field descriptions [141](#)
Check - Modules and Libraries pop-up
 example [144](#)
 field descriptions [144](#)
Check - Preprocessors pop-up
 example [143](#)
 field descriptions [143](#)
CHECK command [43, 97](#)
Check pull-down
 actions [138](#)
 example [138](#)
checking, syntax [139](#)
COBEDIT command [22, 197](#)
COBOL
 changing version [25](#)
 grouping verbs [92](#)
 intelligent search [46, 212](#)
 levels of source statements [92](#)
 OF clause [16](#)
 searching within tree view [70](#)
 specifying version [24](#)
 subsets [12, 51, 121, 153, 213](#)
 using tree view [69](#)
commands
 & Retain [185, 189](#)
 abbreviations [299](#)
 ACTION [89](#)
 ALLOCDEF [139, 191](#)
 BRANCH [95, 192](#)
 BRANCH BACKUP [39](#)
 CHECK [43, 97](#)
 COBEDIT [22, 197](#)
 DATAMAP [199](#)
 diagram syntax [186](#)
 DISPLAY [79, 83, 200](#)
 EDITCOPY [201](#)
 EDITPGM [203](#)
 END [5, 95](#)
 EQUATE [72, 204](#)

- EXCLUDE 17
- EXCLUDE ALL 54
- FINDXTND 17, 212
- HELP 97, 175, 222
- help 307
- HIGH 17, 223
- input area 11, 259
- JUMP 68, 95, 229
- LEVELS 91, 95, 230
- line xi, 97, 183, 283
- LIST 97
- LOCATE 81
- LPRINT 17, 233
- LPUNCH 17, 240
- PARMDEF 246
- PREF 101, 247
- PREPROC 250
- primary xi, 183
- PRINTLOG 251
- PRINTLST 252
- processing 184
- quick reference 299
- RECALL 184, 254
- recalling 254
- REDO 185, 257
- REFRESH 258
- REPEAT 184, 259
- repeating 259, 261
- RESET 95, 260
- RETAIN 185, 189
- RFIND 185, 261
- RHIGH 185, 262
- RPREF 185, 263
- RSCROLL 185, 264
- RTREEVW 185, 265
- RTV 69
- SCROLL 17, 96, 266
- sequence 184
- SET 272
- SETTINGS 6, 10, 35
- SWITCH 274
- tree view 95
- TREEVIEW 275
- UPDATE 277
- ZOOM COPY 279
- ZOOM DEF 280
- ZOOMIN 96, 281
- ZOOMOUT 96, 281
- compiler
 - DBCS flag 195
 - RES/NORES flag 195
- compiler options
 - DATEPROC flag 141
 - YEARWINDOW 141

- concatenating
 - dataname 204
 - multiple datanames 16
- concatenation rules 217
- COND Tree View tag 94
- conventions page xx
- copy library
 - enabling and disabling 28
 - retrieving source 41
- COPY list
 - creating online 28
 - saving and loading 27
 - syntax 28
- Copy/Include libraries 258
- copybook
 - expanding 40
 - removing 42
 - viewing 40
- CUA command 198, 272
 - action bar display 35
 - overview 1
- CUA features x, 5
- cursor position 185, 190
- cursor substitution character 186
- customizing
 - dataset names 167
 - SmartEdit 22

D

- DATACOM Parameter Definition pop-up
 - example 163
 - field descriptions 163
- DATAMAP command 199
- dataname 16, 204
- dataset names, customizing 167
- DB2 Parameter Definition pop-up
 - example 158
 - field descriptions 159
- DB2 support parameter definition 158
- demonstration program 22
- DISPLAY command 79, 83, 200
- displaying action bar 171, 198, 274
- DM (DATAMAP) line command 285
- DMCL name 160

E

- EC (EDITCOPY) line command 286
- Edit - Entry screen 33
- EDITCOPY command 201
- editor parameter
 - INPUT 31
 - IO 32
 - LIBCOPY 32

- NOINPUT 31
- NOIO 32
- NOLIBCOPY 32
- NOOUTPUT 32
- NOPANCOPY 32
- NORETURN 32
- OUTPUT 32
- overriding 31
- PANCOPY 32
- RETURN 32
- editor screen 35
 - controlling format 10, 35
 - example 10
- EDITPGM command 203
- Encore
 - accessing from ESW screen xvi
 - description xiv
- END command 5, 95
- ending tree view 72
- EP (EDITPGM) line command 287
- EQUATE command 72, 150, 204
- equates 148
- errors, locating 45
- Estimate
 - accessing from ESW screen xvi
 - description xiv
- ESW
 - description xii
 - invoking products xv
 - product integration xvi
- EXCLUDE ALL command 54
- EXCLUDE command 17, 88, 103, 206
- excluded lines 54, 290, 293–294
- execution order 62
- F**
- F (First) line command 97, 288
- File - Display Logic Segment Entry pop-up
 - example 79
 - field descriptions 80
- File - Extract COPY List pop-up
 - example 30
 - field descriptions 30
- File - Generated Code Update Facility
 - pop-up
 - example 84
 - field descriptions 84
- File - Load COPY List pop-up
 - example 29
 - field descriptions 29–30
- File - Program Name List pop-up
 - example 82
 - field descriptions 81–82
- File - Pseudo Code Program Selection
 - pop-up
 - example 85
 - field descriptions 85
- File - Save Copy List pop-up
 - example 29
 - field descriptions 30
- File pull-down
 - actions 78
 - example 77
 - in ISPF 4.1 79
 - tree version 79
- files, log/list/punch/work 164
- FINDXTND command 17, 212
- fully-qualified dataname 16
- G**
- Generated Command pop-up 273
- group item 16
- H**
- help
 - commands 177, 222
 - Explanation and Action screen 313
 - for abends 311
 - for commands 307
 - for messages 312
 - for pop-ups 306
 - for screens 306
 - general information 309
 - index 310
 - messages 178
 - navigational commands 305
 - NOTES 308
 - online 303
 - printing messages 313
 - product release 179
 - Table of Contents 309
- Help - About pop-up
 - example 179
 - field descriptions 179
- Help - Specific ASG Command pop-up
 - example 177
 - field descriptions 177
- Help - Specific ASG Message Number
 - pop-up
 - example 178
 - field descriptions 178
- HELP command 8, 11, 97, 175, 222, 253
- Help pull-down
 - actions 176
 - example 175
- HIGH command 12, 17, 45, 223

highlighted set 228
highlighting 217, 223, 260

I

IDMS Load Module Libraries pop-up
 example 161
 field descriptions 161
IDMS Parameter Definition pop-up
 example 159
 field descriptions 160
IDMS support, parameter definition 159
INPUT editor parameter 31
Insight
 accessing from ESW screen xvi
 description xiv
 using analysis functions xvi
intelligent search feature 46
intelligent search function 70
IO editor parameter 32
ISPF 4.1
 displaying action bar 171, 198, 274
 File pull-down 79
 HOME and TAB keys 6
 Settings screen 10, 35
 SWITCH command 274
ISPF Editor action bar, displaying 171, 198, 274

J

J (Jump) line command 97, 229, 289
JUMP command 68, 95, 229

L

L (Last) line command 97, 290
label name 15
LEARN mode 273
LEVELS command 91, 95, 230
LIBCOPY editor parameter 32
Librarian Entry screen 34
LIBRARIAN-INC 41
line commands
 block format 283
 BR (Branch) 284
 DM (DATAMAP) 285
 EC (EDITCOPY) 286
 EP (EDITPGM) 287
 F (First) 97, 288
 J (Jump) 97, 289
 J (jump) 229
 L (Last) 97, 290
 S (Show) 97, 291
 single format 283
 SS (Show Block) 292

SS (Show block) 97
X (Exclude) 97, 293
XX (Exclude Block) 97, 294
ZD (Zoom Def) 295
ZI (Zoom In) 296
ZI (Zoom in) 97
ZO (Zoom Out) 297
ZO (Zoom out) 97

line range 15, 213
List - CALL Statements pop-up
 example 149
 field descriptions 149
List - COBOL Subset Names pop-up 12
 example 153
 field descriptions 153
List - Equates pop-up
 example 150
 field descriptions 150
List - Perform Range Names pop-up
 example 151
 field descriptions 151
List - Program/Subprogram Names pop-up
 example 152
 field descriptions 152
LIST command 97
list file
 customizing dataset name 167
 default options 168
 LPRINT 233
 processing 166
List Menu pop-up 232
List pull-down
 actions 148
 example 148
LOCATE command 81
log file
 default options 168
 processing 166
logic segment 79, 83
logical execution order 62
logical program unit 93
LPRINT command 17, 99, 149, 233
LPUNCH command 17, 240

M

maintenance level 179
message help 312
messages
 long 11, 312
 recalling 254
 short 11, 312
 syntax error 43
MLE compiler option 141
multiple datanames 16

N

naming log/list/punch files 166
 navigating help 305
 NOINPUT editor parameter 31
 NOIO editor parameter 32
 NOLIBCOPY editor parameter 32
 NOOUTPUT editor parameter 32
 NOPANCOPY editor parameter 32
 NORETURN editor parameter 32
 NOTES in help 308

O

online syntax checking 43, 194
 Options - COPY/Include Libraries screen
 actions 25
 example 24
 field descriptions 26
 Options - Define Preprocessor Group pop-up
 example 173
 field descriptions 173
 Options - Equate pop-up
 defining substitution names 72
 example 174
 field descriptions 174
 Options - IDMS Local Mode Dictionaries
 pop-up
 example 162
 field descriptions 162
 Options - Log/List/Punch Definition pop-up
 example 165
 options 166
 Options - Log/List/Punch Name
 Customization pop-up
 example 167
 field descriptions 168
 Options - Preprocessors pop-up
 example 172
 field descriptions 172
 Options - Processing Modes pop-up
 example 169
 field descriptions 170
 Options - Product Allocations pop-up
 example 164
 field descriptions 164
 Options - Product Parameters pop-up
 example 31, 157
 field descriptions 31, 158
 Options pull-down
 actions 156
 example 156
 OUTPUT editor parameter 32

P

PANCOPY editor parameter 32
 Panvalet
 ++INCLUDE 41
 Entry screen 33
 paragraph control transfer methods 101
 PARMDEF command 157, 246
 pattern string 17
 perform range structures 93
 performs 148
 pop-ups
 description 5
 overview 9
 selection fields 9
 Text Fields 9
 PREF command 101, 247
 prefix area 283
 PREPROC command 250
 printing messages 313
 PRINTLOG command 251
 PRINTLST command 252
 product integration xvi
 product level 253
 program
 checking syntax 194
 displaying range 92
 format 91
 hierarchical levels 53, 281
 pseudo code 83
 selecting 85
 structure 53, 281
 VIASMPRT 313
 PROGRAM EXIT statement 35
 programs 148
 pseudo code
 adding lines 83
 updating 278
 pull-down
 Check 137
 description 5
 File 77
 Help 175
 List 148
 Options 156
 overview 8
 Search 108
 View 88
 punch file
 default options 168
 LPUNCH 240
 processing 166

R

RECALL command 184, 254
 recalling commands and messages 254
 Recap
 accessing from ESW screen xvi
 description xiv
 RECUR Tree View tag 94
 REDO command 185, 257
 REFRESH command 24, 139, 258
 release level 253
 release number 179
 REP Tree View tag 94
 REPEAT command 184, 259
 REPEATED Tree View tag 94
 REPETITION option 66
 requesting help 305
 RESET command 88, 95, 103, 139, 260
 RETAIN command 185, 189
 RETURN editor parameter 32
 RFIND command 185, 261
 RHIGH command 185, 262
 RPREF command 185, 263
 RSCROLL command 185, 264
 RTREEVW command 185, 265
 RTV command 69

S

S (Show) line command 97, 291
 screen
 ASG Abend Codes 311
 characteristics 10
 Check - Compiler Options 140
 Check - Modules and Libraries 144
 Check - Preprocessors 143
 DATACOM Parameter
 Definition 163
 DB2 Parameter Definition 158
 description 10
 Edit Entry 33
 editor 35
 File - Display Logic Segment
 Entry 79
 File - Extract COPY List 30
 File - Generated Code Update
 Facility 84
 File - Load COPY List 29
 File - Pseudo Code Program
 Selection 85
 File - Save COPY List 29
 File - Select Logic Segment 82
 format 10
 Generated Command 273
 Help - About 179

Help - Specific ASG Command 177
 Help - Specific ASG Message
 Number 178
 Help Explanation and Action 313
 IDMS Load Module Libraries 161
 IDMS Local Mode Dictionaries 162
 IDMS Parameter Definition 159
 Librarian Entry 34
 List - CALL Statements 149
 List - COBOL Subset Names 12, 153
 List - Equates 150
 List - Perform Range Names 151
 List - Program/Subprogram
 Names 152
 List Menu 232
 messages 11
 Options - COPY/Include Libraries 24
 Options - Define Preprocessor
 Group 173
 Options - Equate 72, 174
 Options - Log/List/Punch
 Definition 165, 233
 Options - Log/List/Punch Name
 Customization 167
 Options - Preprocessors 172
 Options - Processing Modes 169
 Options - Product Allocations 164
 Options - Product Parameters 31, 157
 overview 10
 Panvalet Entry 33
 Search - Any/Unknown Type 129
 Search - Branch Request 132
 Search - COBOL Subset Name 45,
 120
 Search - Data Name 70, 109
 Search - IN-Clause 135
 Search - Label Name 113
 Search - Line Range 126
 Search - Pattern String 116
 Search - Program Name 123
 Selection List 134
 Selection List - Subset Names 52
 SmartEdit Editor 10
 subsets 12
 Tree View 93
 View - Exclude Request 104
 View - Paragraph Cross Reference 60,
 99
 View - Paragraph Cross Reference
 Request 98
 View - Reset Request 103
 View - Scroll Request 106
 View - Tree View Request 63, 91
 SCROLL command 17, 96, 106, 139, 266

- Search - Any/Unknown Type pop-up
 - example 129
 - field descriptions 130
 - in Tree View 132
 - Search - Branch Request pop-up
 - example 132
 - field descriptions 133
 - in Tree View 133
 - Search - COBOL Subset Name pop-up 45
 - example 120
 - field descriptions 121
 - in Tree View 123
 - Search - Data Name pop-up
 - example 70, 109
 - field descriptions 110
 - in Tree View 113
 - Search - IN-Clause pop-up
 - example 135
 - field descriptions 135
 - Search - Label Name pop-up
 - example 113
 - field descriptions 114
 - Search - Line Range pop-up
 - example 126
 - field descriptions 127
 - in Tree View 129
 - Search - Pattern String pop-up
 - example 116
 - field descriptions 117
 - in Tree View 119
 - Search - Program Name pop-up
 - example 123
 - field descriptions 124
 - in Tree View 126
 - search directions 216
 - Search pull-down
 - actions 108
 - COBOL source 108
 - example 108
 - selection fields 9
 - Selection List - Subset Names pop-up 52
 - Selection List pop-ups 134
 - set
 - description 12
 - highlighting 228
 - label name 15
 - line 15
 - SET command 169, 272
 - SETTINGS command 6, 10, 35
 - Settings screen (ISPF 4.1) 10, 35
 - shortened action bar 6
 - SmartDoc
 - accessing from ESW screen xvi
 - description xiv
 - SmartEdit
 - accessing from ESW screen xvi
 - action bar 6
 - activating 23
 - customizing 22
 - description xv
 - introduction 1
 - product level 253
 - SmartQuest
 - accessing from ESW screen xvi
 - description xv
 - SmartTest
 - accessing from ESW screen xvi
 - description xv
 - source manager 24
 - source statements
 - removing 67
 - viewing 66
 - SOURCE Tree View tag 94
 - specifying
 - COBOL language level 24
 - Editor parameters 24
 - editor parameters 31
 - source manager 24
 - SS (Show block) line command 97, 292
 - STOP RUN statement 35
 - subset name 15
 - subsets 12, 121, 148, 153, 213
 - substitution name for character string 72
 - SWITCH command 274
 - switching between editor and tree view 68
 - syntax
 - checking 43, 139
 - error messages 43
 - viewing errors 139
- T**
- TAB key (ISPF 4.1) 6
 - table entry 16
 - tag subsets 14
 - tagged lines subsets 12
 - tags, Tree View screen 94
 - target
 - dataname 15
 - description 15
 - pattern 17
 - screen subsets 14
 - tags 14
 - text fields 9
 - tracing branching logic 35
 - transfer methods 101
 - tree view
 - displaying all levels 93
 - ending 72

Tree View screen
 example [93](#)
 line commands [97](#)
 primary commands [95](#)
 tags [94](#)

Tree View tags
 + (plus) [94](#)
 colon () [94](#)
 COND [94](#)
 RECUR [94](#)
 REPEATED [94](#)
 SOURCE [94](#)

TREEVIEW command [95, 265, 275](#)
troubleshooting [73](#)

U

UPDATE command [277](#)

V

variable, redefined [16](#)
VIAFDEMO program [22](#)
VIASMPRT program [313, 317](#)
View - Exclude Request pop-up
 example [104](#)
 field descriptions [104](#)
View - Paragraph Cross Reference pop-up
 example [60, 99](#)
 field descriptions [61, 100](#)
View - Paragraph Cross Reference Request
 pop-up
 example [98](#)
 field descriptions [98](#)
View - Reset Request pop-up
 example [103](#)
 field descriptions [103](#)
View - Screen Request pop-up
 example [106](#)
 field descriptions [106](#)
View - Tree View Request pop-up
 example [63, 91](#)
 field descriptions [92](#)
View Levels pop-up
 example [105](#)
 field descriptions [105](#)
View pull-down
 actions [89](#)
 actions in Tree mode [90](#)
 example [88](#)
 in Tree View [90](#)
viewing syntax errors [139](#)

X

X (Exclude) line command [293](#)
X (exclude) line command [97](#)
XX (Exclude Block) line command [294](#)
XX (exclude block) line command [97](#)

Y

YEARWINDOW option [141](#)

Z

ZD (Zoom Def) line command [295](#)
ZI (Zoom In) line command [97, 296](#)
ZO (Zoom Out) line command [97, 297](#)
ZOOM COPY command [279](#)
ZOOM DEF command [280](#)
ZOOMIN command [96, 281](#)
ZOOMOUT command [96, 281](#)

